

REFERENTIE-
GIDS

ASHTON · TATE ■TM

© 1985 Ashton-Tate
Reproduction by any method is strictly prohibited

KORTE UITLEG

De dBASE-II-Referentiegid is een aanvulling op de Gebruikershandleiding en het Help-bestand (Help file). Hierin is de informatie die nodig is bij het gebruiken van dBASE II snel te vinden.

CONVENTIES

TYPOGRAFISCHE CONVENTIES

Kleine letters	informatie die de gebruiker in moet geven
HOOFDLETTERS	onderdelen van dBASE II-commando's die letterlijk ingetikt moeten worden
[...]	facultatieve onderdelen van dBASE II-commando's
<...>	onderdelen van dBASE II-commando's die de gebruiker op moet geven

CONVENTIES VOOR DE NAAMGEVING VAN BESTANDEN

Bestandsnamen mogen maximaal 8 tekens (géén spaties) bevatten.

CONVENTIES VOOR DE NAAMGEVING VAN VELDEN

Veldnamen mogen maximaal 10 tekens lang zijn. Ze moeten beginnen met een letter, maar mogen cijfers en ingesloten dubbele punten bevatten. In veldnamen mogen géén spaties opgenomen zijn.

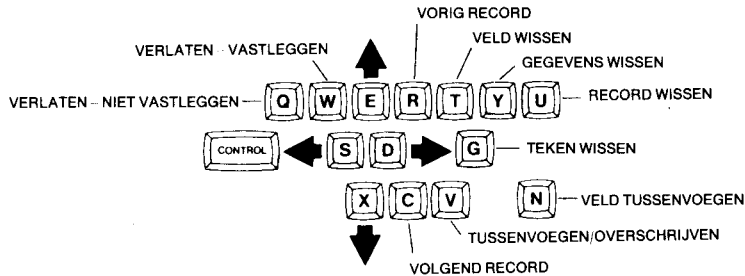
TYPEN DATA IN dBASE II

- C** Gegevens (data) van het type teken ("character"). Omvat alle toetsenbordtekens (ASCII) die afgedrukt kunnen worden.
 - N** Numerieke gegevens voor berekeningen. Omvat getallen, +, - en de decimale punt .
 - L** Logische gegevens. Omvat Y, y, T, t, voor Waar (True), of N, n, F, f voor Onwaar (False)
-

BESTURINGSTOETSEN EN CURSOR-BESTURINGSTOETSEN

“FULL-SCREEN”-BEWERKINGEN (ALLE WERKSTANDEN)

Zie elders voor speciale stuurtekens die in gebruik zijn bij Redigeren (Edit), Wijzigen (Modify), Toevoegen (Append), Aanmaken (Create) of Tussenvoegen (Insert). Gebruik Ctrl-toets met:



TYPEN VAN BESTANDEN IN dBASE II

.**CMD** commandobestand
 .**DBF** databasebestand
 .**FMT** opmaakbestand
 .**FRM** rapportopmaakbestand
 .**MEM** geheugenbestand
 .**NDX** indexbestand
 .**TXT** tekstuitvoerbestand

TERMINOLOGIE

<**begrenzer**> of
 <**scheidingstekens**> elk niet-alfanumeriek teken dat gebruikt wordt om gegevens af te grenzen, b.v. enkele (' ') of dubbele (" ") aanhalingstekens, rechte haken ([]), dubbele punten (: :), komma's (, ,), etc.

<**bereik**> een facultatief onderdeel van het commando dat aangeeft op welke records dBASE II het commando moet uitvoeren. <bereik> heeft drie mogelijke waarden: ALLe records van het bestand; NEXT <n> records; en RECORD <n>. De standaardwaarde is verschillend voor de verschillende commando's.

< commando > of < opdracht >	elk geldig dBASE-commando of geldige functie.
< bestand > of < bestandsnaam >	de naam van een bestand dat aangemaakt of geopend moet worden.
< geheugenvar > of < gehvar >	de naam van een geheugenvariabele.
< indexbestand >	de naam van een indexbestand dat aangemaakt of geopend moet worden.
< lijst geheugenvar >	een lijst door komma's gescheiden geheugenvariabelen.
< lijst uitdr >	een lijst door komma's gescheiden uitdrukkingen.
< lijst velden >	een lijst door komma's gescheiden veldnamen.
< n >	een getal dat dBASE II als een constante gebruikt.
< raamwerk >	voor bewerkingen op meerdere bestanden, mits van hetzelfde type en/of met eenzelfde tekenreeks in de naam; maakt sequentiële ("batch")bewerkingen mogelijk op groepen geheugenvariabelen met eenzelfde tekenreeks in de naam.
< rekenkundige uitdr >	een uitdrukking waarvan de inhoud Numeriek (N) moet zijn.
< sleutel >	het onderdeel of de onderdelen van een databasebestand in gebruik om een indexbestand aan te maken.
< tekenreeks >	tekenreeks(en). Gewoonlijk moeten ze tussen enkele (' ') of dubbele (" ") aanhalingstekens of rechte haken ([]) staan.
< tekenreeks-uitdr >	een < uitdr > die een waarde van het type Teken (C) op moet leveren. Dat kan een tekenreeks, een geheugenvariabele of een veld met een inhoud van het type C zijn of elke combinatie die daaruit met een operator is opgebouwd.
< uitdr > of < uitdrukking >	een item of reeks items en operatoren waarvan dBASE II de waarde kan bepalen. Afhankelijk van het type van de gegevens die ze bevatten kunnen uitdrukkingen C, N of L zijn.
< veld >	de naam van een veld in een record.

OPERATOREN voor het verwerken van gegevens.

LOGISCHE OPERATOREN (op volgorde van prioriteit)

- () haakjes voor groeperen
- .NOT.** logisch "niet" (eenwaardige operator)
- .AND.** logisch "en"
- .OR.** logisch "of"
- \$** logische operator voor deelreeksen



REKENKUNDIGE OPERATOREN (op volgorde van prioriteit)

- () haakjes voor groeperen
- *
- /
- +
-

RELATIONELE OPERATOREN

- <
- >
- =
- <>
- <=
- >=



REEKSOPERATOREN

- +
-



FUNCTIES

* de gewist-record-functie: geeft logisch Waar (T) als resultaat indien bij het huidige record een wismarkering staat.

de recordwijzer (een geheugenvariabele) waarin het nummer van het huidige record is opgeslagen.

& macrofunctie die in een commandoreeks een waarde gebruikt die in een geheugenvariabele is opgeslagen (de geheugenvariabele moet van het type 'tekenreeks' (C) zijn).

!(**<tekenreeks-uitdr>**) de hoofdletterfunctie: zet kleine letters om in hoofdletters.

\$(<uitdr>, <begin>, <lengte>) de deelreeksfunctie: geeft dat deel van de <uitdr> dat bepaald wordt door <begin> en <lengte>.

@(**<tekenreeks-uitdr 1>**, **<tekenreeks-uitdr 2>**) de deelreeks-zoekfunctie: levert een geheel getal op dat de plaats aangeeft waar de eerste reeks voorkomt binnen de tweede.

CHR(<rekenkundige uitdr>) geeft het ASCII-teken dat overeenkomt met de waarde die de uitdrukking oplevert.

DATE () de naam van de systeemvariabele die de systeemdatum bevat.

EOF de functie End-Of-File: levert logisch Waar (T) op indien het laatste record van het open bestand gepasseerd is.

FILE (<bestand>) de functie FILE geeft logisch Waar (T) als het opgegeven bestand op de aangesloten schijf aanwezig is.

INT (<rekenkundige uitdr>) de geheel-getal-functie: kapt alles dat rechts van de decimale punt staat af om de waarde die de uitdrukking oplevert om te zetten in een geheel getal.

LEN (<tekenreeks-uitdr>) de lengtefunctie: geeft op uit hoeveel tekens de gespecificeerde tekenreeks bestaat.

RANK (<tekenreeks-uitdr>) levert de ASCII-waarde op van het meest linkse teken van de <tekenreeks-uitdr>.

STR (<rekenkundige uitdr>, <lengte>, [decimalen >]) de reeksfunctie: zet het gespecificeerde deel van de rekenkundige uitdrukking om in een tekenreeks.

TRIM (<tekenreeks-uitdr>) de trimfunctie: verwijdert spaties die aan het eind van de opgegeven tekenreeks zitten.

TYPE (<uitdr>) de typefunctie: geeft het type van de opgegeven uitdrukking.

VAL (<tekenreeks-uitdr>) de waardefunctie: zet een tekenreeks (die moet bestaan uit cijfers) om in een rekenkundige uitdrukking.

dBASE II-COMMANDO'S

? <uitdr> bepaalt de waarde van een uitdrukking en geeft die weer.

@ <coördinaten> [SAY <uitdr> [USING 'beeld']] [GET <var> [PICTURE 'beeld']] opmaak van uitvoer naar scherm of afdrukeenheid.

ACCEPT ['prompt'] TO <geheugenvar> vraagt de gebruiker een tekenreeks in te voeren die dan opgeslagen wordt in de opgegeven geheugenvariabele. Bij de invoer zijn geen aanhalingstekens nodig.

APPEND [BLANK] voegt een of meer records of lege geformatteerde records toe aan het databasebestand.

APPEND FROM <bestand> [SDF] of [DELIMITED] [FOR <uitdr>] voegt gegevens uit een databasebestand of System Data Formatbestand toe aan het open databasebestand.

BROWSE [FIELDS <lijst velden>] laat het open databasebestand zien en maakt wijzigingen over het volledige scherm mogelijk.

CALL <geheugenvar> een machinetaal-commando.

CANCEL breekt uitvoer van een programmabestand af.

CHANGE [<bereik>] FIELD <lijst> [FOR <uitdr>] wijzigt opgegeven velden van geselecteerde records.

CLEAR herstelt de begintoestand van dBASE II.

CONTINUE zet de uitvoering van het LOCATE-commando voort.

COPY TO <bestand> STRUCTURE [FIELD <lijst>] kopieert de structuur van het open databasebestand naar het opgegeven bestand.

COPY TO <bestand> [<bereik>] [FIELD <lijst>] [FOR <uitdr>] kopieert het open databasebestand naar het opgegeven bestand.

COPY TO <bestandsnaam> [<bereik>] [FIELD <lijst>] [SDF] [DELIMITED [WITH <begrenzer>]] [FOR <uitdr>] kopieert gegevens van het open databasebestand naar het bestand met de opgegeven <bestandsnaam>.

COPY TO <bestand> STRUCTURE EXTENDED maakt een databasebestand aan waarvan de records de zelfde indeling hebben als het open bestand.

COUNT [<bereik>] [FOR <uitdr>] [TO <geheugenvar>] telt de records die aan een bepaalde voorwaarde voldoen. Zonder opgave van <bereik> worden alle records geteld.

CREATE [<bestand>] begint het aanmaken van een nieuw bestand.

CREATE <nieuw bestand> FROM <oud bestand> EXTENDED maakt een nieuw databasebestand aan waarvan de structuur bepaald wordt door de gegevens die opgeslagen zijn in het oude bestand. (Wordt samen met COPY STRUCTURE EXTENDED gebruikt.)

DELETE [<bereik>] [FOR <uitdr>] markeert de aangegeven records voor verwijdering.

DELETE FILE <bestand> wist het genoemde bestand.

DISPLAY [<bereik>] [<lijst velden>] [FOR <uitdr>] [OFF] laat records van het open databasebestand zien.

DISPLAY FILES [ON <disk drive>] [LIKE <raamwerk>] laat zien welke bestanden op de schijf staan. Standaard betreft dat de databasebestanden op de aangesloten schijf.

DISPLAY MEMORY laat de geldende geheugenvariabelen zien.

DISPLAY STATUS laat informatie over het systeem zien: o.a. de datum, open bestanden, indexbestanden en sleutels, en alle SET-parameters.

DISPLAY STRUCTURE maakt de structuur van het open bestand zichtbaar.

EDIT [n] voor het selectief wijzigen van het open databasebestand.

EJECT geeft de afdrukeenheid opdracht het papier verder te draaien tot het begin van een volgende bladzijde.

ERASE wist het beeldscherm.

FIND <tekenreeks> zoekt in geïndexeerde databasebestanden het eerste record met een sleutel die met de tekenreeks overeenkomt.

GO of GOTO <n> of <TOP> of <BOTTOM> gaat naar het opgegeven record of de opgegeven plaats in het open databasebestand.

HELP [<commando of ander sleutelwoord>] geeft een overzicht van het Help-bestand of maakt informatie zichtbaar die over een bepaald commando, een bepaalde functie, e.d., in het Help-bestand is opgenomen.

INDEX ON <sleutel> **TO** <bestand> stelt een indexbestand op voor het open databasebestand.

INPUT ['prompt'] **TO** <geheugenvar> vraagt de gebruiker om een <uitdrukking> in te voeren die opgeslagen wordt in een bestaande geheugenvariabele.

INSERT [BEFORE] [BLANK] voegt een nieuw record in een databasebestand in.

JOIN TO <bestand> **FOR** <uitdr> [FIELDS <lijst>] maakt een nieuw databasebestand op uit die records uit (binnen primair en secundair werkgebied open staande) bestanden, waarvoor de <uitdr> Waar is.

LIST [<bereik>] [lijst velden] [FOR <uitdr>] [OFF] laat de records uit het open bestand zien.

LIST FILES [ON <disk drive>] [LIKE <raamwerk>] laat zien welke bestanden op de schijf staan. Standaard betreft dat databasebestanden op de aangesloten schijf.

LIST MEMORY laat de geldende geheugenvariabelen zien.

LIST STATUS laat informatie over het systeem zien: o.a. de datum, open bestanden, de indexbestanden en -sleutels, en alle SET-parameters.

LIST STRUCTURE laat de structuur van het open bestand zien.

LOCATE [< bereik >] [FOR < uitdr >] zoekt het eerste record op dat voldoet aan de voorwaarde. Gebruik CONTINUE om het volgende record te bereiken dat ook aan de voorwaarde voldoet.

MODIFY COMMAND < bestand > wijzigt het opgegeven commandobestand.

MODIFY STRUCTURE wijzigt de structuur van het open databasebestand.

NOTE of * zorgt ervoor dat dBASE II de rest van de commandoregel niet leest. Wordt gebruikt om opmerkingen in het commandobestand op te kunnen nemen.

PACK wist records met een wismarkering.

QUIT [TO < lijst CP/M-commando's of .COM-bestanden >] beëindigt het werken met dBASE II en geeft de besturing terug aan het besturingssysteem. Facultatief kunnen andere programma's aangeroepen worden.

READ wordt gebruikt na een "full-screen"-bewerking van een ingedeeld beeldscherm, en aan het begin van het inlezen van gegevens via GET-commando's.

RECALL [< bereik >] [FOR < uitdr >] heft de wismarkeringen op.

REINDEX werkt de indexbestanden bij die niet open waren tijdens het wijzigen van het corresponderende databasebestand.

RELEASE [< lijst geheugenvariabelen >] of [ALL] of [ALL LIKE < raamwerk >] of [ALL EXCEPT < raamwerk >] verwijdert ongewenste geheugenvariabelen uit het geheugen.

REMARK geeft de er op volgende tekens weer. Wordt in commandobestanden gebruikt om opmerkingen af te beelden op beeldscherm of afdrukeenheid.

RENAME < oud bestand > TO < nieuw bestand > verandert de naam van een bestand.

REPLACE [< bereik >] < veld > WITH < uitdr > [, < veld2 > WITH < uitdr2 > ...] [FOR < uitdr >] vervangt de inhoud van de gespecificeerde velden of records door de opgegeven waarden.

REPORT [FORM < bestand >] [< bereik >] [FOR < uitdr >] [TO PRINT] stelt een rapportopmaakbestand op of gebruikt een bestaand opmaakbestand voor uitvoer van gegevens in een door de gebruiker opgegeven opmaak.

RESET [< drive >] herstelt CP/M na wisselen van schijven.

RESTORE FROM < bestand > [ADDITIVE] haalt geheugenvariabelen die opgeslagen waren in een .MEM-bestand op en activeert ze. Geldende geheugenvariabelen worden gewist tenzij het facultatieve 'ADDITIVE' wordt gebruikt.

RETURN beëindigt de uitvoering van een programmabestand en geeft de besturing terug aan het aanroepende bestand of gaat terug naar de interactieve toestand.

SAVE TO <bestand> [ALL LIKE <raamwerk>] of [ALL EXCEPT <raamwerk>] slaat de (gespecificeerde) geldende geheugenvariabelen op in een .MEM-bestand.

SELECT [PRIMARY] of [SECONDARY] schakelt om tussen het primaire en het secundaire werkgebied van dBASE II.

SET zie bij SET-COMMANDO'S.

SKIP [+/- n] springt het aangegeven aantal records voor- of achteruit. Standaardwaarden zijn +1 en +n.

SORT ON <sleutel> TO <file> [ASCENDING] of [DESCENDING] maakt een versie van het open databasebestand aan waarin de records alfabetisch of numeriek gerangschikt zijn naar de inhoud van het als sorteersleutel opgegeven veld. Standaard wordt gesorteerd in opklimmende volgorde.

STORE <uitdr> TO <geheugenvar> berekent de waarde van de uitdrukking en slaat die waarde op in de aangegeven geheugenvariabele.

SUM <lijst velden> [TO <lijst geheugenvariabelen>] [<bereik>] [FOR <uitdr>] berekent en toont de som van numerieke velden.

TOTAL TO <bestand> ON <sleutel> [FIELDS <lijst velden>] [FOR <uitdr>] maakt een samenvatting van het open, geïndexeerde of gesorteerde databasebestand, bestaande uit totalen van de numerieke velden van records die dezelfde sleutel bevatten.

UPDATE FROM <bestandsnaam> ON <sleutel> [ADD <lijst velden>] of [REPLACE <lijst velden>] werkt het open bestand groepsgewijs bij met behulp van gegevens uit het FROM-bestand.

USE <bestand> [INDEX <lijst bestanden>] opent een databasebestand, en facultatief de gewenste indexbestanden.

WAIT [TO <geheugenvar>] onderbreekt de uitvoering van dBASE II tot er een teken vanaf het toetsenbord wordt ingevoerd; facultatief wordt dat teken opgeslagen in een geheugenvariabele.

SET-COMMANDO'S

De standaardwaarde voor elk SET-commando wordt aangegeven door de volgorde van de waarden: OFF/ON geeft aan dat de standaardwaarde OFF is; bij ON/OFF is dat ON.

SET ALTERNATE TO [<bestand>] maakt het aangegeven bestand aan (van het type .TXT) om uitvoer naar het scherm op te slaan op schijf. Herhaling zonder <bestand> sluit het bestand.

SET ALTERNATE OFF/ON ON slaat uitvoer naar het scherm op in het bestand dat aangemaakt is met het vorige commando; OFF stopt het opslaan.

SET BELL ON/OFF ON laat een zoemtoon horen bij het invoeren van ongediende gegevens of overschrijden van een veldgrens; OFF zet de zoemer af.

SET CARRY OFF/ON ON kopieert bij gebruik van APPEND gegevens uit het laatste record naar een nieuw record; OFF schakelt die functie uit.

SET COLON ON/OFF ON geeft met dubbele punten de lengte van in te voeren variabelen aan; OFF onderdrukt de dubbele punten.

SET CONFIRM OFF/ON ON wacht op indrukken van de <return>-toets alvorens naar het volgende veld gesprongen wordt; met OFF wordt naar het volgende veld gesprongen zodra het huidige veld vol is.

SET CONSOLE ON/OFF ON maakt alle uitvoer op het scherm zichtbaar; OFF onderdrukt de uitvoer naar het scherm.

SET DATE TO <DD/MM/JJ> stelt de systeemdatum in of bij.

SET DEBUG OFF/ON ON stuurt alle uitvoer van SET ECHO en STEP naar de afdrukeenheid; OFF stuurt die uitvoer naar het scherm.

SET DEFAULT TO <drive> zorgt ervoor dat dBASE II de opgegeven drive voor verdere handelingen als standaarddrive gebruikt.

SET DELETED OFF/ON Bij ON kan dBASE II bij commando's waaraan een bereik meegegeven kan worden records met een wismarkering niet lezen of bewerken; bij OFF kan dBASE II alle records lezen.

SET ECHO OFF/ON Bij ON worden alle commando's bij uitvoering van een commandobestand op het scherm zichtbaar gemaakt; OFF onderdrukt weergave van de commando's.

SET EJECT ON/OFF Bij ON geeft REPORT een papierdoorvoer tot het begin van de volgende bladzijde alvorens een rapport wordt afgedrukt; bij OFF gebeurt dat niet.

SET ESCAPE ON/OFF ON geeft de gebruiker de mogelijkheid uitvoering van een commandobestand te onderbreken door aanslaan van de <esc>-toets; OFF schakelt de werking van de <esc>-toets uit.

SET EXACT OFF/ON ON vereist exacte overeenstemming bij het vergelijken van twee reeksen (bijvoorbeeld bij gebruik van FOR <uitdr> of FIND); bij OFF kunnen ook reeksen van verschillende lengte met elkaar overeenkomen.

SET FORMAT TO [<bestand>] maakt bij uitvoeren van APPEND, EDIT, INSERT, etc., gebruik van een opmaakbestand (type .FMT) voor weergave op scherm of afdrukeenheid.

SET FORMAT TO <scherm/afdrukeenheid> stuurt de uitvoer van het opmaakbestand naar het beeldscherm of naar de afdrukeenheid.

SET HEADING TO <tekenreeks> slaat de tekenreeks op en geeft die weer als kopregel bij gebruik van REPORT. (Tekensreeks maximaal 60 tekens lang.)

SET INDEX TO [<lijst bestanden>] installeert indexbestanden voor gebruik bij het corresponderende databasebestand. Het eerstgenoemde indexbestand wordt de actieve index; alle andere .NDX-bestanden worden automatisch bijgewerkt bij veranderen van het databasebestand.

SET INTENSITY ON/OFF ON maakt gebruik mogelijk van omgekeerd beeld of dubbele intensiteit bij "full-screen"-bewerkingen (mits de apparatuur dit toelaat); OFF schakelt een dergelijke weergave uit.

SET LINKAGE OFF/ON ON verplaatst de recordwijzers van zowel primair als secundair werkgebied bij commando's die de wijzer alleen naar achteren verplaatsen (bijvoorbeeld commando's met een <bereik>); OFF schakelt die koppeling tussen de wijzers uit.

SET MARGIN TO <n> stelt de linkerkantlijn van de afdrukeenheid in op <n> spaties.

SET PRINT OFF/ON ON stuurt uitvoer naar de afdrukeenheid; OFF onderbreekt uitvoer naar de afdrukeenheid.

SET RAW OFF/ON ON geeft bij LIST en DISPLAY records weer zonder spaties tussen de velden; OFF voegt een extra spatie in tussen de velden.

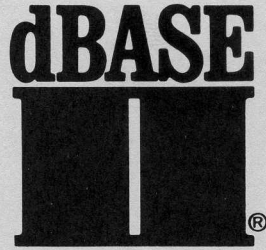
SET SCREEN ON/OFF ON maakt "full-screen"-bewerkingen mogelijk bij APPEND, EDIT, CREATE etc.; OFF schakelt die mogelijkheid uit.

SET STEP OFF/ON ON onderbreekt uitvoering van een commandobestand na elk commando — waardoor fouten opgespoord kunnen worden; OFF schakelt dat onderbreken uit.

SET TALK ON/OFF ON geeft de resultaten van commando's op het scherm weer; OFF onderdrukt weergave van de resultaten.

COMMANDO'S VOOR GESTRUCTUREERD PROGRAMMEREN

DO <bestand >	opent een commandobestand (type .CMD) en voert het uit.
DO CASE CASE <uitdr1 > <opdrachten > CASE <uitdr2 > <opdrachten > CASE <uitdrN > <opdrachten > OTHERWISE <opdrachten > ENDCASE	voert één (en niet meer dan één) van verschillende mogelijke programmaroutes (CASE <uitdr1 > t.e.m. CASE <uitdrN >) uit. Levert CASE <uitdr > logisch Waar (T) op voor die uitdrukking, dan worden de bijbehorende commando's uitgevoerd. Facultatief kan OTHERWISE voor een alternatieve route opgenomen worden. Een DO CASE-structuur moet afgesloten worden met ENDCASE.
DO WHILE <uitdr > <opdrachten > LOOP <opdrachten > ENDDO	opent een gestructureerde commandolus. De eropvolgende commando's en opdrachten worden uitgevoerd totdat de uitdrukking logisch Onwaar (F) oplevert. Door LOOP, een facultatief onderdeel van de lus, wordt teruggesprongen en de uitdrukking opnieuw bekeken. Een DO WHILE-structuur moet afgesloten worden met ENDDO.
IF <uitdr > <opdrachten > ELSE <opdrachten > ENDIF	maakt voorwaardelijke uitvoering van opdrachten en commando's mogelijk. ELSE (facultatief) geeft een alternatieve route. Een IF structuur moet afgesloten worden met ENDIF.
TEXT <tekst > ENDTEXT	maakt het mogelijk tekst weer te geven binnen een commandobestand. dBASE II vat alles als weer te geven tekst op tot het commando ENDTEXT bereikt wordt.



**Relationeel
Database Management
System
(geschreven in assembler)**

ASHTON · TATE ■TM

Postbus 71876, 1008 EB Amsterdam
Telefoon 020-462515

© 1985 Ashton-Tate
Reproduction by any method is strictly prohibited

dBASE II®
HANDLEIDING VOOR DE GEBRUIKER

Deze handleiding is en blijft eigendom van Ashton-Tate. De handleiding bevat vertrouwelijke informatie. De houder van deze handleiding verplicht zich aan derden inzage te onthouden. Het computerprogramma dBASE II en deze handleiding worden beschermd door de auteurswet (waaronder begrepen eventuele persoonlijkheidsrechten) en de overige wetgeving met betrekking tot industriële eigendom.

**BELANGRIJKE INFORMATIE BETREFFENDE HET GEBRUIK
VAN dBASE II EN VAN DEZE HANDLEIDING**

Het is niet toegestaan om het computerprogramma dBASE II en/of de daarbij behorende handleiding op enige wijze te vermenigvuldigen, openbaar te maken, over te brengen op een elektronische gegevensdrager of op andere wijze in een op machinaal leesbare vorm om te zetten zonder uitdrukkelijke en schriftelijke toestemming van Ashton-Tate B.V.

Het zonder toestemming van Ashton-Tate B.V. vermenigvuldigen van het computerprogramma en/of de daarbij behorende handleiding is op grond van de auteurswet (waaronder begrepen eventuele persoonlijkheidsrechten) en overige wetgeving met betrekking tot bescherming van industriële eigendom, verboden. Bij overtreding van betreffende bepalingen van de wet zal door Ashton-Tate B.V. steeds in een civiele procedure schadevergoeding worden gevorderd en zal tevens aangifte worden gedaan bij de Officier van Justitie waarop een strafvervolgning van overtreders kan plaatsvinden.

In een door de gebruiker en namens Ashton-Tate B.V. ondertekende software-licentie-overeenkomst zijn rechten en plichten van partijen vastgelegd.

Deze handleiding is bijgewerkt en aangevuld tot de dag van verzending en wordt in die vorm aan de gebruiker verstrekt.

Behalve de garantie(s) die in de software-licentie-overeenkomst zijn opgenomen wordt door Ashton-Tate B.V. geen enkele andere garantie gegeven.

Ashton-Tate B.V. verleent uitdrukkelijk geen garantie voor de verkoopbaarheid van het produkt of voor de geschiktheid voor een bepaald doel.

Ashton-Tate B.V. zal in geen geval aansprakelijk zijn voor enige directe of indirecte vorm van schade of verlies door een afnemer geleden, zelfs niet wanneer de afnemer Ashton-Tate B.V. van te voren op de hoogte heeft gesteld dat schade of verlies zal gaan ontstaan. (Voorzover beperking of uitsluiting van schade door de wet is beperkt of uitgesloten, is deze bepaling niet van toepassing.)

INHOUD

1.0	dBASE gebruiken	1
2.0	Systeemvereisten	4
3.0	dBASE-bestanden	5
3.1	Database-bestanden	5
3.2	Geheugenbestanden	6
3.3	Commandobestanden	7
3.4	Rapportbestanden	7
3.5	Tekstuitvoerbestanden	8
3.6	Indexbestanden	8
3.7	Formatbestanden	8
4.0	Uitdrukkingen	9
4.1	Functies	10
4.2	Bewerkingen	16
5.0	Macrosubstitutie	19
6.0	Koppeling aan niet-dBASE processors	20
7.0	Soorten commando's	21
8.0	Werken met volledig beeldscherm	24
9.0	Commando's	26
9.1	Symbooldefinities	26
9.2	Gulden regels	28
10.0	Machinetaal-interface	31
10.1	Geheugenkaart	31
10.2	Machinetaal-commando's	32
Bijlagen	A. Commando's	149
	B. Limietwaarden en systeembeperkingen	153
	C. Foutmeldingen	155

1.0 dBASE GEBRUIKEN

Om het dBASE-programma te kunnen gebruiken, plaatst u de dBASE distributiediskette (bij voorkeur een kopie van die diskette) in een vrije drive. Deze drive kiest u nu als standaard-drive (b.v. als de diskette in drive "B" wordt geplaatst, typt u "B:" en dan < return >) en daarna typt u het volgende:

dBASE

Het programma wordt dan in het geheugen geladen en bij 8-bits computers wordt U allereerst verzocht de datum in te vullen:

ENTER DATE AS DD/MM/JJ OR RETURN FOR NONE:

Deze datum wordt vermeld op elke database die tijdens de komende sessie gewijzigd wordt en in geprinte RAPPORT-titels van rapporten die tijdens deze sessie zijn opgesteld. De geldigheid van de datum wordt gecontroleerd.

OPGELET: Deze controle geldt niet voor 29 februari van de jaren 1900 en 2100. Om de getallen van elkaar te scheiden, kunt u een schuine streep of een ander speciaal teken (behalve de punt) gebruiken.

Voorbeelden van geldige data:

1,1,81
02 02 02
17/3/83

Bij 16-bits computers verschijnt vervolgens het aanmeldingsbericht:

*** dBASE II VER 2.xxx***

De punt op de tweede regel is de dBASE-prompt die aanduidt dat dBASE gereed is voor uw commando's. Commando's voor dBASE staan meestal in de gebiedende wijs: een werkwoord, eventueel gevolgd door zinsneden die de bewerking nader bepalen. Voordat er iets uitgevoerd wordt, gaat dBASE eerst de hele regel na. Als dBASE een fout in het commando ontdekt, wordt de gebruiker op de hoogte gesteld door middel van foutmeldingen op de console. Gewoonlijk kan de gebruiker de foutieve commando's corrigeren en er veranderingen in aanbrengen, zodat niet het hele commando opnieuw ingevoerd hoeft te worden. Als dBASE een fout ontdekt die niet precies kan worden beschreven, gaat hij ervan uit dat dit een syntaxisfout is; in zo'n geval wordt de foutieve regel op het scherm herhaald, met een vraagteken boven het deel dat de verwarring veroorzaakte.

2 ... dBASE

Voorbeelden van herstel:

.DISPRAY MEMORY *** UNKNOWN COMMAND DISPRAY MEMORY CORRECT AND RETRY? Y CHANGE FROM :PR CHANGE TO :PL DISPLAY MEMORY MORE CORRECTIONS? < return > .STORE (2 + 2 TO X *** SYNTAX ERROR *** ? STORE (2 + 2 TO X CORRECT AND RETRY? Y CHANGE FROM : + 2 CHANGE TO : + 2) STORE (2 + 2) TO X MORE CORRECTIONS? N 4	foutief commando herhaald ja, corrigeren verander de letters PR in PL na de verandering < return > = geen veranderingen meer de reeks (2 + 2 wordt aangewezen nee, geen veranderingen meer het resultaat
.SUM TO X NO EXPRESSION TO SUM SUM TO X CORRECT AND RETRY? N	uitleg geen verandering, breek dit commando af

Een programma kan ook op de volgende manier uitgevoerd worden:

DBASE < bestand >

Hierdoor wordt dBASE in het geheugen opgeslagen; er wordt een commandobestand < bestandsnaam > geopend en er wordt een begin gemaakt met de uitvoering van het commandobestand. Deze manier is met name interessant als de dBASE-applicatie onderdeel moet uitmaken van een omvangrijker systeem, zoals bijvoorbeeld in de kettingoptie ("chaining") in het dBASE QUIT-commando.

BESTURINGSTEKENS

- < control > -P – Schakelt printer aan/uit (zie ook commando SET PRINT)
- < control > -U – Wist huidige record.
- < control > -Y – Wist huidige regel (behalve bij full-screen editing)
- < rubout > – Wist laatst ingevoerde teken
- < control > -H (of spatie-terug-toets) – Wist laatst ingevoerde teken
- < escape > – Om uit bepaalde mogelijk lang lopende commando's te komen: bijv. DISPLAY, COUNT, DELETE, INPUT, LIST, LOCATE, RECALL, REPLACE, SKIP, en SUM.
< escape > dient ook om ACCEPT, INPUT, REPORT (interactief) en WAIT te verlaten. In alle gevallen wordt met < escape > de besturing teruggebracht naar de interactieve monitor en op het scherm verschijnt de dBASE-prompt (een punt).

Bij het uitvoeren van een commando-bestand controleert dBASE bij het begin van elke nieuwe commandoregel of < escape > is ingedrukt.

Opmerking: Met het commando SET ESCAPE OFF kunt u < escape > vergrendelen.

4 ... VEREISTEN

2.0 SYSTEEMVEREISTEN

Om dBASE op een efficiënte manier te kunnen gebruiken, moet u beschikken over een systeem dat beschikt over:

- I. Voor een op een 8080 of Z-80 gebaseerd microprocessor-systeem:
 - a) 48K bytes (of meer) geheugen, met CP/M (dBASE gebruikt geheugen tot A400 hex.) Opmerking: Voor bepaalde apparatuur, zoals Apple, Heath en Northstar, is door een te grote CP/M-module een geheugen van meer dan 48K vereist;
 - b) CP/M-besturingssysteem versie 2.x of recentier;
 - c) Eén of meerdere geheugeneenheden onder CP/M besturing (meestal floppy- of Winchester-drives);
 - d) Een met cursor bestuurbaar beeldscherm (bij voorkeur 24 regels en 80 kolommen) voor de full-screen bewerkingen;
 - e) Eventueel een tekstprinter (voor bepaalde commando's).

- II. Voor een op een 8086 of 8088 gebaseerd microprocessor-systeem:
 - a) 128K bytes (of meer) geheugen, incl. besturingssysteem;
 - b) CP/M-86 of MS-DOS/PC-DOS besturingssysteem;
 - c) Eén of meerdere geheugeneenheden;
 - d) Een met cursor bestuurbaar beeldscherm (bij voorkeur 24 regels en 80 kolommen) voor full-screen bewerkingen;
 - e) Eventueel een tekstprinter (voor bepaalde commando's).

3.0 dBASE-BESTANDEN

Een bestand (file) is in feite een verzameling gegevens op een geheugeneenheid die door de gebruiker zijn ingevoerd. De informatie kan in het bestand worden opgeslagen of eruit worden opgevraagd. Bestanden kunnen in zes groepen worden onderverdeeld, één voor elke specifieke bewerking in dBASE.

Alle dBASE-bestanden zijn standaard CP/M- of MS-DOS-bestanden met een naamveld van 8 tekens en een bestandstype van 3 tekens. Hieronder ziet u een lijst van de standaard bestandstypes die door dBASE gebruikt worden. Voor een commando dat toegang tot het bestand geeft, hoeft het suffix niet te worden opgegeven; dBASE zal ervan uitgaan dat het standaardtype bedoeld wordt. Als er b.v. reeds een bestand bestaat met als type DBF, dan hoeft u in de commando's voor verwerking van dit bestand dit niet te specificeren.

DATABASE FILES (databasebestanden)	- .DBF
MEMORY FILES (geheugenbestanden)	- .MEM
COMMAND FILES (commandobestanden)	- .CMD (of .PRG voor 16-bits versie)
REPORT FORM FILES (rapportbestanden)	- .FRM
TEXT OUTPUT FILES (tekstuitvoerbestanden)	- .TXT
INDEX FILES (indexbestanden)	- .NDX
FORMAT FILES (formatbestanden)	- .FMT

Om naar een dBASE-bestand te verwijzen moet u een geldige CP/M-bestandsnaam gebruiken. Vergeet niet dat dBASE van de bovengenoemde types uitgaat als u bij de toegang tot de bestanden het type niet opgeeft. Voor meer informatie in verband met het gebruik van bestandsnamen en -types verwijzen we naar de publicatie van Digital Research "CP/M User's Guide".

3.1 DATABASE-BESTANDEN (.DBF) (Data Base Files)

Databases vormen de kern van dBASE. dBASE-gegevensbankbestanden bestaan uit een structuurrecord en maximaal 65535 gegevensrecords. Het structuurrecord is in feite een overzicht van het format van het gegevensrecord. De structuur kan uit maximaal 32 verschillende componenten bestaan. Elke component in de structuur verwijst naar een veld met gegevens in de gegevens-records. De structuur bevat de volgende gegevens:

- * De naam van de gegevensvelden
- * Het gegevensstype in de gegevensvelden
- * De grootte van de gegevensvelden
- * De plaats van de gegevens in de records

NAAM GEGEVENSVELD – De naam mag maximaal 10 tekens lang zijn. Voor alle handelingen tijdens de uitvoering van dBASE wordt met deze naam naar de gegevensvelden verwezen. Veldnamen zijn alfanumeriek (plus dubbele punten). De veldnamen moeten echter met een letter beginnen en dubbele punten mogen niet aan het begin of het eind staan. Hieronder enkele voorbeelden.

6 ... BESTANDEN

Voorbeelden van namen van gegevensvelden:

A
A123456789
ABC:DEF
A:B:C:D:E
ABCD: ongeldig, dubbele punt niet binnen naam
ABC,DEF ongeldig, komma niet toegestaan

GEGEVENSTYPE – In dBASE bestaan drie gegevenstypes om de inhoud van gegevensvelden te specificeren: de tekenreeks ('ABCD'), numerieke hoeveelheden (2 of 5*18), en logische waarden (waar/niet waar).

VELDGROOTTE – Hiermee wordt het aantal tekenposities (lengte) bedoeld dat nodig is om gegevens in een bepaald veld te kunnen plaatsen. Tekenreeksen en numerieke velden mogen 1 tot 254 posities lang zijn. Bij een numeriek veld moet ook het decimaalteken worden meegerekend. Logische velden zijn altijd 1 positie lang. Bij numerieke velden kan ook het aantal decimale cijfers rechts van het decimaalteken in de structuur worden opgenomen.

Zodra de structuur is vastgelegd, kunt u voor elk gewenst aantal records gegevens in de velden invoeren. Meestal is er slechts één gestructureerd gegevensbestand tegelijk beschikbaar (dit wordt het USE-bestand of het bestand in USE genoemd). U kunt echter ook twee gegevensbestanden tegelijk gebruiken. Zie commando's SELECT en JOIN.

3.2 GEHEUGENBESTANDEN (.MEM) (MEMory files)

Geheugenbestanden zijn statische bestanden in een geheugen, die net als records in variabelen verdeeld zijn. Deze variabelen noemen we geheugenvariabelen; u kunt maximaal beschikken over 64 van dergelijke variabelen.

De waarden van de geheugenvariabelen zijn afhankelijk van de database die gebruikt wordt. Met andere woorden, de recordpositie van het bestand in USE heeft geen invloed op de variabelen in het geheugenbestand. Geheugenvariabelen bestaan uit constanten, uitkomsten van berekeningen, symbolische substitutiereeksen (zie Hoofdstuk 5), enz. De regels voor het bepalen van de naam, het type en de grootte van de geheugenvariabelen zijn dezelfde als voor de veldvariabelen hierboven.

Met het commando SAVE (bewaar) worden alle huidige geheugenvariabelen in een geheugenbestand geschreven; met het commando RESTORE (geef terug) wordt een opgeslagen geheugenbestand opnieuw in het geheugen gelezen.

3.3 COMMANDOBESTANDEN (.CMD) (.PRG voor 16-bits versie)

In een commandobestand staat een reeks dBASE-commando's. Hiermee kunt u een aantal vaak gebruikte commandoreeksen (in feite programma's) opslaan, wat de bewerking van de bestanden vereenvoudigt en versnelt.

Commandobestanden kunnen met text editors en/of tekstverwerkers aangemaakt en gewijzigd worden. dBASE biedt echter dank zij het commando `MODIFY COMMAND` ("wijzig commando"), ook zélf mogelijkheden commandobestanden te creëren en te wijzigen. Commandobestanden worden met het `DO`-commando gestart. In een commandobestand kunnen alle dBASE-commando's voorkomen; voorzichtigheid is hier geboden, omdat bepaalde commando's (`CREATE`, `INSERT`, `APPEND` (vanaf het toetsenbord)) naast de inhoud van het commandobestand en van andere bestanden, ook invoer van de gebruiker verwachten.

Commandobestanden mogen genest worden, d.w.z. commandobestanden mogen `DO`-commando's bevatten, die op hun beurt commandobestanden aanroepen. Ook hier is voorzichtigheid geboden, omdat dBASE maximaal 16 bestanden tegelijk kan openen. Als er dus een bestand in `USE` is, kan tot max. 15 commando-bestanden diep genest worden. Bepaalde commando's gebruiken ook werkbestanden (b.v. `SORT` gebruikt 2 extra bestanden; `REPORT`, `INSERT`, `COPY`, `SAVE`, `RESTORE` en `PACK` gebruiken elk één extra bestand.) Als u bijvoorbeeld een `SORT` commando gebruikt in het laagste commandobestand binnen de nesting, dan kunt u slechts 13 commandobestanden gebruiken (m.a.w. het `USE`-bestand, 2 `SORT`-werkbestanden en 13 commandobestanden = 16). Wanneer een commandobestand het `RETURN`-commando bevat of wanneer de uitvoering van een commandobestand ten einde is, dan wordt dit commandobestand gesloten en komt de ingenomen plaats vrij voor een ander commandobestand.

3.4 RAPPORTBESTANDEN (.FRM) (report FoRM files)

Met het commando `REPORT` wordt ofwel een nieuw formulierbestand aangemaakt, ofwel een bestaand formulierbestand gebruikt. In het formulierbestand staan instructies voor de rapportgenerator met betrekking tot titels, aanheffen, kolominhouden en voor de berekening van totalen. dBASE bouwt binnen de `REPORT`-dialoog formulierbestanden op, die gewijzigd kunnen worden met text-editors of tekstverwerkers. Dit is evenwel een mogelijke bron voor fouten en wordt, behalve voor gevorderde dBASE-gebruikers, afgeraden.

8 ... *BESTANDEN*

3.5 **TEKSTUITVOERBESTANDEN (.TXT) (TeXT output files)**

De tekstuitvoerbestanden worden gecreëerd als u de commando's "SET ALTERNATE TO < bestandsnaam >" en "SET ALTERNATE TO ON" geeft. Zie SET-commando's voor meer informatie. Ook met de commando's COPY en APPEND wordt een tekstbestand (.TXT) gecreëerd zodra u de opties SDF (System Data Format) of DELIMITED gebruikt.

3.6 **INDEXBESTANDEN (.NDX) (INDeX files)**

Met het INDEX-commando van dBASE worden indexbestanden gemaakt. Daarin zitten sleutels en pointers (wijzers) naar records van een database. Indexeren is een dBASE-techniek waarmee gegevens in een grote database snel gelokaliseerd kunnen worden. Zie het INDEX-commando voor meer informatie.

3.7 **FORMATBESTANDEN (.FMT) (ForMaT files)**

In een formatbestand zitten alleen "@"-opdrachten en "*" -commentaren. Een dergelijk bestand wordt gespecificeerd door het commando "SET FORMAT TO < bestandsnaam >" en wordt in werking gesteld door de daaropvolgende READ-commando's. Net zoals commandobestanden (die veel gelijkenis vertonen met formatbestanden) kunnen formatbestanden gecreëerd en gewijzigd worden met een goede tekstverwerker of met de MODIFY COMMAND-functie. Format-bestanden zijn echter niet noodzakelijk. "@" en "*" -instructies zullen meestal aanwezig zijn in de commandobestanden waarin ze nodig zijn.

4.0 UITDRUKKINGEN

Een uitdrukking in dBASE is een groep enkelvoudige items en operatoren die zodanig geëvalueerd kunnen worden dat ze een nieuwe enkelvoudige waarde vormen. Voorbeeld: "2 + 2" is een uitdrukking die kan worden geëvalueerd tot de waarde "4". Uitdrukkingen zijn niet zonder meer altijd numeriek. De uitdrukking 'abc' + 'def' kan geëvalueerd worden tot de waarde 'abcdef' (aaneenschakeling van tekenreeksen), en de uitdrukking 1 > 2 als de logische (Booleaanse) waarde. '.F.' (false – niet-waar).

Uitdrukkingen in dBASE worden opgebouwd uit de volgende elementen:

- * Veldvariabelen in een database
- * Geheugenvariabelen
- * Constanten binnen de commando's (zgn. 'literals')
- * Functies
- * Bewerkingen

VARIABLEN – Een variabele is in dBASE een willekeurig gegevensveld waarvan de waarde kan veranderen. Er zijn twee soorten variabelen: namen van gegevensvelden (de gegevens in het veld van een dBASE database-record kunnen gewijzigd worden wanneer de database verplaatst of gewijzigd wordt); en namen van geheugenvariabelen (waarin informatie staat die wordt gewijzigd als een commando STORE, RESTORE, COUNT, SUM, WAIT, ACCEPT, of INPUT, etc. wordt gegeven).

Er zijn drie soorten variabelen:

- * Tekenreeksen (character strings)
- * Numerieke hoeveelheden
- * Logische waarden

CONSTANTEN – Een constante of 'literal' is een gegevensitem met een onveranderlijke, zelf-bepaalde waarde: 1, 'abc' en .T. zijn constanten met een constante waarde, ongeacht de positie van de database of van een commando voor geheugenvariabelen. Het zijn constanten omdat ze zelf de waarde ZIJN die ze voorstellen (in tegenstelling tot variabelen, dwz. namen die een waarde herbergen). De waarden die ze voorstellen zijn, resp.: een numerieke één, een tekenreeks (met de letters "a", "b" en "c") en een logische (Booleaanse) waarde TRUE (".T." – waar).

Tekenreeks-constanten moeten tussen enkele aanhalingstekens ('), dubbele aanhalingstekens (") of rechte haakjes ([,]) staan. Als een tekenreeks één van deze "scheidingstekens" zelf moet bevatten, dan moet dat teken tussen een paar andere scheidingstekens staan. Voorbeeld: de reeksen 'abc[def]ghi' en [abc'def'ghi] zijn geldige karakterreeksen, terwijl 'abc'def'ghi' dat niet is.

Logische constanten (waar/niet-waar) worden voorgesteld door "T", "t", "Y" of "y" voor 'waar' (true of yes) en "F", "f", "N" of "n" voor 'niet-waar' (false of no).

10 ... FUNCTIES

4.1 FUNCTIES

Functies zijn bijzondere bewerkingen die u in uitdrukkingen kunt gebruiken om iets uit te voeren dat met de gewone uitdrukkingen moeilijk of onmogelijk is. In dBASE zijn er drie soorten functies: numerieke, teken- en logische functies. Die onderscheiding is gebaseerd op de soort waarden die de functie oplevert.

Geheel getal-functie

INT(< numerieke uitdrukking >)

Met deze functie wordt een numerieke uitdrukking geëvalueerd waarbij de waarden na het decimaalteken genegeerd worden zodat een geheel getal ontstaat.

Voorbeelden:

```
. ? INT (123.456)
123
. STORE 123.456 TO X
123.456
. ? INT (X)
123
```

Recordnummer-functie:

#

De waarde van de recordnummer-functie is het gehele getal dat overeenkomt met het nummer van het huidige record.

Voorbeelden:

```
. ? #
4
. SKIP
. ? #
5
```

(we gaan er vanuit dat een database "in USE" is en op recordnummer 4 staat)

Reeks-functie (STRing):

STR(< numerieke uitdrukking > , < lengte > [, < decimalen >])

Met deze functie wordt een numerieke uitdrukking geëvalueerd, hetgeen een tekenreeks oplevert. De waarde van de STR-functie is een tekenreeks met lengte < lengte > . Als er < decimalen > opgegeven zijn, wordt hiermee het aantal cijfers achter het decimaalteken bedoeld. De argumenten kunnen constanten, variabelen of uitdrukkingen zijn.

Pas op: Als u deze functie gebruikt om een sleutel voor het indexeren te genereren, MOETEN de argumenten constanten zijn.

Voorbeeld:

```
. ? STR(123.456,9,3)
123.456
```

Deelreeks-functie (substring):

```
$( < tekenuitdrukking > , < start > , < lengte > )
```

Deze functie levert een tekenreeks op uit een gespecificeerd gedeelte van een andere reeks. De waarde van de subreeks-functie is een tekenreeks met lengte < lengte > en met tekens uit de tekenuitdrukking, beginnend bij het tekennummer < start > voor < lengte > tekens. < start > en < lengte > kunnen constanten, variabelen of uitdrukkingen zijn.

Indien < lengte > langer is dan de < tekenuitdrukking > of de < tekenuitdrukking > tussen < lengte > en < start > niet genoeg tekens heeft, dan worden alleen de tekens gegeven die aanwezig zijn. Zie de volgende voorbeelden.

Pas op: Als u deze functie gebruikt om een sleutel voor het indexeren te genereren, MOETEN de argumenten constanten zijn.

Voorbeelden:

```
. ? $('abcdefghi',3,3)
cde
. store 3 TO m
3
. store 3 TO n
3
. ? $('abcdefghi',m,n)
cde
. ? $('abcdefghi',6,7)
fghi
. DISPLAY FOR '8080'$TITLE
```


12 ... FUNCTIES

Reeks-naar-Numeriek-functie

VAL(< tekenreeks >)

Met deze functie wordt een geheel getal gevormd op basis van een tekenreeks, bestaande uit cijfers, tekens en max. één decimaalteken. De lengte van het geheel getal komt overeen met het aantal tekens in de reeks. Als de reeks met cijfers begint, maar ook niet-numerieke tekens bevat, dan komt de waarde die met de VAL-functie berekend wordt overeen met de vooropstaande numerieke tekens.

Een andere manier om cijferreeksen (-strings) om te zetten in numerieke waarden is het gebruik van "&" (zie 5.0 Macro's). "&" zet de reeks in een numerieke waarde om (ook het decimaalgedeelte) waar deze substitutie wordt aangetroffen.

Voorbeelden:

```
. ? VAL('123')
123
. ? VAL ('123xxx')
123
. ? VAL ('123.456')
123
. STORE '123.456' TO NUM
123.456
. ? 14 + &NUM
137.456
```

Rang-functie

RANK (< tekenreeks >)

Met deze functie verkrijgt u de gehele waarde (of ASCII-waarde) van het eerste teken in de tekenreeks < tekenreeks >. Deze functie komt overeen met de ASC-functie in de meeste BASIC-versies.

Voorbeeld:

```
. ? RANK('A')
65
```

Lengte-functie

LEN(< tekenreeks >)

Met deze functie verkrijgt u een geheel getal met als waarde het aantal tekens in de genoemde reeks.

Voorbeeld:

```
. STORE 'abc' TO STRING
. ? LEN(STRING)
3
```

Gewist Record-functie

*

Dit is een logische functie die .TRUE. (waar) is als het record een wismarkering gekregen heeft en .FALSE. (niet-waar) als dit niet zo is.

Voorbeeld:

```
. ? *
.T. (verondersteld dat een database in USE is en het
record met het DELETE-commando een
wismarkering heeft gekregen)
```

End-of-File-functie

EOF

Dit is een logische functie die .TRUE. (waar) is als het einde van het bestand in USE bereikt is (het huidige record is dan het laatste record in de database).

Voorbeelden:

```
. ? EOF
.F. (verondersteld dat een database in USE
. GOTO BOTTOM is die niet op het laatste record staat).
. ? EOF
.F.
. SKIP
. ? EOF
.T.
```

Zoekfuctie voor deelreeksen

@ (< tekenreeks 1 > , < tekenreeks 2 >)

Deze functie levert een geheel getal op waarvan de waarde overeenkomt met de positie in < tekenreeks 2 > , welke het begin is van een deelreeks gelijk aan < tekenreeks 1 > . Als reeks 1 niet in reeks 2 voorkomt, dan is de waarde van de @-functie nul. Opmerking: De @-functie lijkt op de deelreeks operator "\$"; alleen wordt hier verteld waar de eerste reeks gevonden kan worden binnen de tweede; dit kan worden uitgedrukt als: "waar zit reeks 1 in reeks 2".

14 ... FUNCTIES

Voorbeeld:

```
. ? @ ('def', 'abcdefghi')  
4
```

Hoofdletter-functie

```
!( < tekenreeks-uitdrukking > )
```

Met deze functie verkrijgt u dezelfde reeks als de tekenreeks-uitdrukking, met dit verschil dat de kleine letters worden omgezet in hoofdletters.

Voorbeeld:

```
. ? !('abc')  
ABC
```

Getal-naar-Teken-functie

```
CHR( < numerieke uitdrukking > )
```

Deze functie geeft het equivalent in ASCII-tekens van de numerieke uitdrukking. Als de uitdrukking b.v. het getal 13 is, genereert CHR(13) volgens ASCII een < return >. Deze functie is nuttig als de gebruiker rechtstreeks hardware-eenheden, meestal printers, moet besturen.

Voorbeeld:

```
. ? 'abcd' + CHR(13) + ' ____ ' (zou op de printer abcd onderstrepen)  
abcd
```

Datum-functie

```
DATE( )
```

Deze functie levert een tekenreeks op, waarin de systeemdatum in de vorm XX/XX/XX gegeven wordt. De tekenreeks bestaat altijd uit 8 tekens. Tussen de haakjes wordt niets ingevoerd, ze duiden er alleen op dat het hier om een functie gaat (om verwarring te vermijden met variabelen met de naam "DATE").

Met het commando SET DATE TO kunt u de dBASE-systeemdatum invoeren bij het opstarten van dBASE of op een ander moment. Bij het opstarten wordt de datum op geldigheid gecontroleerd (zie format-optie in het INSTALL-programma). Met SET DATE TO < reeks > (geen aanhalings-tekens) wordt de datum niet gecontroleerd, zodat u een datum kunt invoeren in de vorm MM/DD/JJ, DD/MM/JJ of JJ/MM/DD.

Voorbeelden:

```
. ? DATE()
06/15/81
. STORE DATE() TO gehvar
06/15/81
. SET DATE TO 4 1 82
. ? DATE()
04/01/82
```

Bestand-functie

FILE(< reeksuitdrukking >)

Dit is een logische functie die .TRUE. (waar) is als er een bestand onder de naam < reeksuitdrukking > bestaat en .FALSE. (niet-waar) als deze niet bestaat.

Voorbeeld:

```
. USE ONTWERP
. ? FILE('ONTWERP')
.T.
```

Type-functie

TYPE(< uitdrukking >)

Deze functie levert een reeks van één teken op dat het type van de < uitdrukking > aangeeft; een "C" voor Character (teken), een "N" voor Numeriek, een "L" voor Logisch, en "U" voor Undefined (niet-gedefinieerd).

Voorbeeld:

```
. STORE 1 TO X
. ? TYPE(X)
N
```

Trim-functie

TRIM(< tekenreeks >)

Met de TRIM-functie worden spaties aan het einde van een veld verwijderd. Meestal worden alle variabelen in dBASE gevolgd door spaties om problemen te vermijden bij het uitlijnen van kolommen op het scherm.

Opmerking: Deze functie mag NIET in het INDEX-commando gebruikt worden omdat de sleutellengte voor intern dBASE-gebruik berekend moet kunnen worden.

16 ... BEWERKINGEN

Voorbeelden:

```
. STORE 'ABC ' TO S
. ? LEN(S)
6
. STORE TRIM(S) TO S
. ? LEN(S)
3
```

4.2 BEWERKINGEN

Er zijn vier soorten bewerkingen: rekenkundige, vergelijkende, logische en reeksbewerkingen. Hieronder volgen de specifieke operatoren voor de verschillende bewerkingen en voorbeelden voor de minder bekende operatoren.

Vergeet niet dat de gegevens aan beide zijden van de operator van hetzelfde type moeten zijn. U kunt dus alleen gehele getallen bij gehele getallen optellen, of tekens bij tekens samenvoegen; een geheel getal bij een teken optellen interpreteert dBASE als een syntaxisfout.

```
. STORE 3 TO A
3
. STORE '3' TO B
3
. ? A + B
*** SYNTAX ERROR ***
?
? A + B
CORRECT AND RETRY(Y/N)?
```

Deze fout ontstaat doordat numerieke waarden en tekens door de computer anders geïnterpreteerd worden; een numerieke waarde 3 is gewoon 3 (3 hex), terwijl een teken 3 de ASCII-waarde van 33 hex heeft. Het programma raakt in de war, weet niet of er een optelling of samenvoeging bedoeld wordt. Laten we dezelfde variabelen als in het vorige voorbeeld gebruiken:

```
. ? A + VAL(B)
6
```

De reeks '3' is in een geheel getal omgezet en de optelling is uitgevoerd.

REKENKUNDIGE OPERATOREN (geven rekenkundige uitkomsten)

+ = optelling
- = aftrekking
* = vermenigvuldiging
/ = deling
() = haakjes voor rekenkundige prioriteit

Voorbeelden:

- . ? (4 + 2) * 3 Een voorbeeld van het gebruik van haakjes
18 voor het aangeven van de rekenkundige
. ? 4 + (2 * 3) prioriteit in berekeningen
10

VERGELIJKINGSOPERATOREN (geven een logische waarde)

- < = kleiner dan
- > = groter dan
- = = gelijk aan
- < > of # = niet gelijk aan
- < = = kleiner dan of gelijk aan
- > = = groter dan of gelijk aan
- \$ = deelreeks-operator (als A en B tekenreeksen zijn, dan is A\$B
alleen waar als reeks A gelijk is aan B, of A in B aanwezig is.

Voorbeelden:

- . ? 'abc'\$'abcdefghi' Voorbeeld van deelreeks-operator \$
.T.
- . ? 'abcd'\$'ghijkl'
.F.
- . DISPLAY FOR '8080'\$TITEL Brengt alle records op het scherm
 waarbij ergens in het veld TITEL
 '8080' staat. *'Sapnek'*

LOGISCHE OPERATOREN (geven een logische uitkomst)

- .OR. = Booleaans OF
- .AND. = Booleaans EN
- .NOT. = Booleaans NIET (eenwaardige operator)

Voorbeelden:

- . STORE T TO A
.T.
- . STORE F TO B
.F.
- . ? A .OR. B
.T.
- . STORE .NOT. B TO C
.T.
- . ? A .AND. C
.T.

18 ... VOLGORDE

REEKS-OPERATOREN (geven reeksuitkomsten)

- + = reeksaaneenschakeling ('concatenatie')
- = reeksaaneenschakeling waarbij spaties naar rechts verplaatst worden

Voorbeelden:

. STORE 'ABCD ' TO A
ABCD
. STORE 'EFGH' TO B
EFGH
. ? A + B
ABCD EFGH
. STORE 'ABCDE ' TO A
ABCDE
. STORE '1234 67' TO B
1234 67
. ? A-B
ABCDE1234 67

In een reeksaaneenschakeling worden de twee reeksen gewoon samengevoegd

In een reeksaaneenschakeling met spatieverschuiving (-) worden de spaties achteraan ('trailing blanks') naar het einde van de reeks verplaatst.

Spaties vooraan en tussen tekens worden niet verplaatst.

Opmerking: A-B en A + B hebben dezelfde lengte.

VOLGORDE VAN UITVOERING

De operatoren worden in een bepaalde volgorde behandeld. Welke berekening wordt vóór welke andere berekening uitgevoerd? In onderstaande tabel staat de volgorde aangegeven voor de drie types operatoren. Binnen elk niveau (1, 2, enz.) gaat de uitvoering van links naar rechts.

Voorbeeld:

. ?? 4 + 2*3
10

Prioriteit rekenkundige operatoren	Prioriteit reeks-operatoren	Logische operator
1) haakjes, functies	haakjes, functies	.NOT.
2) eenwaardige +,-	vergelijking, deelreeks-operator \$.AND.
3) *,/	+,- (aaneenschakeling)	.OR.
4) +,-		
5) vergelijking		

5.0 MACRO-SUBSTITUTIE

Als het systeem in een commando een & tegenkomt, gevolgd door de naam van een tekenreeks-geheugenvariabele, vervangt dBASE het &-teken en de naam van de geheugenvariabele door de tekenreeks die is opgeslagen in de geheugenvariabele. Hierdoor kunt u bepaalde delen van een commando één keer definiëren en deze zo vaak u wilt in diverse commando's oproepen.

Macro's zijn interessant als ingewikkelde uitdrukkingen vaak gebruikt moeten worden. Hierdoor wordt het ook mogelijk parameters door te geven in nesten van commandobestanden. Alle tekens tussen & en het volgende bijzondere teken (zoals een spatie) worden beschouwd als de naam van de geheugenvariabele.

Als u tekens aan de symbolische substitutie wilt toevoegen, dan moet de naam van de geheugenvariabele afgesloten worden met een punt. Bij het vervangen wordt de punt tegelijk met het &-teken verwijderd.

Als op een &-teken geen voor een geheugenvariabele geldige naam volgt, dan wordt er niets vervangen en blijft het &-teken in de commando-regel staan.

Voorbeelden:

```
. ACCEPT "Voer letter voor datadisk-drive in" to DR
. STORE DR + 'DATAFILE' TO DR
USE &DR                                     (bij de uitvoering wordt dit:
USE B:DATAFILE als "B:" ingevoerd
werd als antwoord op ACCEPT)
```

```
. STORE 'DELETE RECORD' TO T
&T 5                                       (Bij de uitvoering wordt dit:
DELETE RECORD 5)
```


20 ... KOPPELING

6.0 KOPPELING AAN NIET-dBASE PROCESSORS

dBASE kan gegevens uit bestanden lezen die gecreëerd zijn met programmatuur die geen gebruik maakt van dBASE (b.v. BASIC, FORTRAN, PASCAL); bovendien kan dBASE bestanden creëren die door andere programmatuur gelezen kunnen worden.

Met het APPEND-commando kan het systeem tekstbestanden lezen in standaard-ASCII (met de CP/M-conventie van een tekstregel, gevolgd door een return en een regelsprong) door de SDF- (Systeemgegevensformat-) optie op te geven. Zo kunt u met het commando COPY eveneens standaard ASCII-format bestanden creëren, wanneer u de SDF-optie gebruikt. Tenzij expliciet anders bepaald, zijn de bestanden die met de opties SDF en DELIMITED gecreëerd worden, .TXT-bestanden.

Met bepaalde processors en talen worden bestanden in een beperkt format geschreven en gelezen. In deze vorm worden alle velden gescheiden door komma's en staan de tekenreeksen tussen aanhalingstekens. Als het sleutelwoord DELIMITED in het commando staat, kan dBASE deze bestanden toevoegen (APPEND) of kopiëren (COPY). SDF wordt verondersteld als u de DELIMITED-mogelijkheid gebruikt.

Aangezien bepaalde processors enkele, en andere dubbele aanhalingstekens gebruiken om tekenreeksen aan te geven, kunt u met APPEND beide mogelijkheden gebruiken. Met het COPY-commando worden gewoonlijk enkele aanhalingstekens gegeven, maar elk willekeurig teken dat in de WITH-uitdrukking van de DELIMITED-bepaling staat, kan worden weergegeven. Wij raden u ten zeerste aan alleen enkele of dubbele aanhalingstekens te gebruiken.

Een bijzonder geval doet zich voor wanneer een “,” in de WITH-uitdrukking gebruikt wordt voor een COPY. Alle spaties aan het einde van tekenreeksen en aan het begin van numerieke waarden, worden verwijderd. Bovendien worden tekenreeksen niet tussen aanhalingstekens of andere tekens gezet.

Voorbeelden:

```
.USE <bestandsnaam> .DBF  
.COPY TO <bestandsnaam> .TXT DELIMITED WITH''
```

```
.USE <bestandsnaam> .DBF  
.APPEND FROM <bestandsnaam> .DAT SDF
```

7.0 SOORTEN COMMANDO'S

In gewoon gebruik van dBASE worden verschillende commando's in bepaalde combinaties gebruikt om een bepaalde taak te vervullen. Zulke groepen worden hieronder besproken. Bepaalde dBASE-commando's zijn opgebouwd volgens de gestructureerde benadering die de meeste "moderne" computertalen gebruiken. Deze commando's staan in de groep COMMANDO-BESTAND-commando's. Een aantal speciale regels bepalen het gebruik van deze commando's die in hoofdstuk 9.0 verder uitgelegd worden.

BESTANDEN CREËREN – met de volgende commando's worden database-bestanden en daarbij behorende bestanden gecreëerd:

- *CREATE – creëert nieuwe gestructureerde database-bestanden
- *COPY – kopieert bestaande databases
- *MODIFY – verandert de structuur van databases
- *REPORT – creëert een rapportbestand
- *SAVE – kopieert de geheugenvariabelen op een geheugen-eenheid
- *INDEX ON – creëert een index-bestand
- *REINDEX – rangschikt opnieuw een oud index-bestand
- *JOIN – geeft de samenvoeging van twee databases
- *TOTAL – geeft een database van records waarvoor het totaal werd berekend.

GEGEVENS TOEVOEGEN – met de volgende commando's worden nieuwe gegevens-records toegevoegd aan databases:

- *APPEND – voegt gegevens toe aan het einde van het bestand
- *CREATE – maakt toevoeging van gegevens mogelijk bij aanmaken/creëren
- *INSERT – last gegevens in een bestand in

GEGEVENS WIJZIGEN – met de volgende commando's worden de gegevens binnen een database gewijzigd:

- *CHANGE – wijzigt kolommen met velden
- *BROWSE – bekijkt en wijzigt records met gebruik van het volledige scherm als venster
- *DELETE – geeft records een wismarkering
- *EDIT – wijzigt bepaalde gegevensvelden in een database
- *PACK – verwijdert records met wismarkering
- *RECALL – schrapt de wismarkering
- *REPLACE – vervangt gegevensvelden door waarden
- *READ – vervangt gegevens vanaf het door de gebruiker gevulde scherm
- *UPDATE – maakt batch-revisie van een database mogelijk

HULP-COMMANDO'S – met de volgende commando's kan de gebruiker on-line informatie opvragen

- *DISPLAY STATUS – geeft informatie over het gegevensbestand en systeem-parameters
- *DISPLAY FILES – toont bestanden van de in gebruik zijnde disk-drive
- *HELP – geeft uitleg over dBASE-commando's en andere informatie

22 ... COMMANDO'S

WEERGAVE VAN GEGEVENS – met de volgende commando's worden bepaalde gegevens uit een database op het scherm gebracht:

- *@ – geeft weergave op scherm of printer van door de gebruiker geformatteerde gegevens
- *BROWSE – geeft weergave van max. 19 records, met zoveel velden als op het scherm passen (browse ≈ “grasduinen”)
- *COUNT – telt het aantal records die aan een bepaalde voorwaarde voldoen
- *DISPLAY – geeft weergave van records, velden en uitdrukkingen
- *READ – geeft weergave van gegevens en prompt-informatie met gebruik van het volledige beeldscherm
- *REPORT – formatteert en geeft weergave gegevensrapport
- *SUM – berekent en geeft weergave van de som van een uitdrukking voor een groep database-records
- *? – geeft weergave van een lijst met uitdrukkingen
- *TEXT – geeft weergave van een blok tekstgegevens

PLAATSINGS-COMMANDO'S – met de volgende commando's wordt de pointer voor het record op een bepaald record gepositioneerd:

- *CONTINUE – positioneert op volgend record onder voorwaarden die in het LOCATE-commando bepaald zijn
- *FIND – positioneert op record dat overeenkomt met een sleutel, voor geïndexeerde bestanden
- *GOTO – positioneert op een bepaald record
- *LOCATE – zoekt een record dat aan een bepaalde voorwaarde voldoet
- *SKIP – positioneert voor- of achteruit

BEWERKING BESTANDEN – met de volgende commando's worden hele database-bestanden beïnvloed

- *APPEND – voegt records toe in dBASE-bestanden of bestanden in System Data Format (SDF)
- *COPY – kopieert databases naar andere databases of SDF-bestanden
- *DELETE – wist bestanden
- *DO – voor het opgeven van een commandobestand waaruit de verdere commando's moeten worden genomen
- *RENAME – geeft een bestand een nieuwe naam
- *SELECT – schakelt om tussen twee bestanden in gebruik (USE)
- *SORT – maakt een kopie van een database die gesorteerd wordt op basis van één van de gegevensvelden
- *USE – bepaalt het database-bestand dat voor alle bewerkingen gebruikt moet worden totdat een nieuwe USE gegeven wordt

GEHEUGENVARIABLEN-COMMANDO'S – met de volgende commando's worden geheugenvariabelen bewerkt

- *ACCEPT – slaat een tekenreeks op in geheugenvariabelen
- *COUNT – slaat optellingen op in geheugenvariabelen
- *DISPLAY – kan geheugenvariabelen weergeven
- *INPUT – slaat uitdrukkingen op in geheugenvariabelen
- *RESTORE – zoekt verzamelingen opgeslagen geheugenvariabelen op
- *SAVE – bewaart de geheugenvariabelen in een bestand
- *STORE – slaat resultaten van uitdrukkingen op in geheugenvariabelen
- *SUM – slaat resultaten van sommingen op in geheugenvariabelen
- *WAIT – neemt één enkele toetsaanslag op in een geheugenvariabele

COMMANDO-BESTAND-COMMANDO'S – de volgende commando's dienen als hulp bij het opbouwen en gebruiken van commandobestanden:

- *ACCEPT – maakt het mogelijk tekenreeksen in geheugenvariabelen in te voeren
- *CANCEL – annuleert de uitvoering van het commandobestand
- *DO – zorgt voor uitvoering van commandobestanden en maakt gestructureerde lussen (loops) in commandobestanden mogelijk
- *IF – zorgt voor voorwaardelijke uitvoering van commando's
- *ELSE – kiest een andere weg binnen een IF
- *ENDDO – eindwoord voor DO WHILE-commando
- *ENDIF – eindwoord voor IF-commando
- *INPUT – maakt invoer van uitdrukkingen in geheugenvariabelen mogelijk
- *LOOP – gaat verder (SKIPt) naar het begin van een DO WHILE-commando-structuur
- *MODIFY COMMAND – maakt het mogelijk commandobestanden te wijzigen
- *RETURN – beëindigt een commandobestand
- *SET – stelt dBASE-besturingsparameters in
- *WAIT – onderbreekt tijdelijk de verwerking van het commandobestand

APPARATUUR-COMMANDO'S – met de volgende commando's wordt randapparatuur zoals printers en beeldschermen bestuurd:

- *EJECT – start een nieuwe pagina bij het lijsten
- *ERASE – maakt het beeldscherm schoon

8.0 WERKEN MET VOLLEDIG BEELDSCHERM (FULL SCREEN)

Hier volgen de cursortoetsen voor werking over het gehele scherm:

- control-E,A – Keert terug naar het vorige gegevensveld.
- control-X,F – Gaat vooruit naar het volgende gegevensveld.
- control-S – Gaat een teken terug in een gegevensveld.
- control-D – Gaat een teken vooruit in een gegevensveld.
- control-Y – Overschrijft de rest van het huidige veld met spaties.
- control-V – Schakelt om tussen 'overschrijven' en 'inlassen'.
- control-G – Wist het teken waarop de cursor staat.
- <rubout> – Wist het teken links van de cursor.
- control-Q – Verlaat de volledige-beeldscherm-verwerking en keert terug naar gewone dBASE besturing. Wijzigingen van database-variabelen worden genegeerd.

In EDIT:

- control-U – Schakelt om tussen aan- of afwezigheid van een wismarkering.
- control-R – Schrijft het huidige record terug op de schijf en brengt het vorige record op het scherm, m.a.w. gaat één record terug.
- control-C – Schrijft het huidige record terug naar schijf en brengt het volgende record op het scherm, m.a.w. gaat één record vooruit.
- control-W – Schrijft het huidige record terug naar schijf en verlaat de volledige-beeldscherm-EDIT-modus (ctl-O voor Superbrain).

In MODIFY:

- control-N – Schuift alle regels één positie lager, zodat een nieuwe regel kan worden ingelast.
- control-T – Wist de cursorregel en plaatst alle regels daaronder één regel hoger.
- control-C – Rolt pagina omlaag.
- control-R – Rolt regel omhoog; rolt pagina naar boven wanneer cursor bovenaan het scherm staat.
- control-W – Schrijft gegevens over op de schijf en gaat verder met normale werkingen (ctl-O voor Superbrain).
- control-Q – Verlaat zonder de wijzigingen te bewaren.

In APPEND, CREATE of INSERT:

- control-C of control-R – Schrijft lopend record over naar de schijf en gaat naar volgende record.
- < return > – als er geen wijzigingen werden aangebracht en de cursor zich op de eerste positie bevindt: afsluiten van de bewerking en terugkeren naar normale dBASE-werking.

in BROWSE:

- control-U – Schakelt om tussen aan- of afwezigheid van een wismarkering.
- control-R – Schrijft het huidige record terug naar schijf en gaat naar het vorige record op het scherm, m.a.w. gaat één record terug.
- control-C – Schrijft het huidige record terug naar schijf en gaat naar het volgende record op het scherm, m.a.w. gaat één record vooruit.
- control-W – Schrijft het huidige record terug naar schijf en verlaat de volledige-beeldscherm-modus (ctl-O voor Superbrain).
- control-Z – Verschuift het venster één veld naar links.
- control-B – Verschuift het venster één veld naar rechts.

9.0 COMMANDO'S

Dit hoofdstuk geeft de expliciete definities van de dBASE-commando's. Voordat u de rest van de commando-informatie doorleest is het echter raadzaam eerst volledig vertrouwd te zijn met de informatie die hieraan voorafgaat.

9.1 SYMBOOLDEFINITIES

Het is zeer belangrijk de betekenis van de speciale symbolen in de algemene verschijningsvorm van de dBASE-commando's goed tot u door te laten dringen. Behalve de vorm van het commando, zullen u hierdoor ook de mogelijkheden van elk commando duidelijk worden. Lees de volgende lijst grondig door.

< commando's > of < opdrachten >

Alle geldige dBASE-opdrachten; betekent ook *volledige* opdrachten. IF zonder ENDIF (of DO zonder ENDDO) is slechts een halve opdracht, terwijl REPORT op zichzelf al een volledige opdracht is.

< tekenreeks >

Een willekeurige tekenreeks; tekenreeksen zijn tekens die tussen enkele aanhalingstekens ('), dubbele aanhalingstekens (") of rechte haakjes ([]) staan.

< scheidingstekens > of < begrenzer >

Een speciaal teken; speciale tekens zijn tekens op het toetsenbord die (in tegenstelling tot cijfers en letters) als leestekens beschouwd kunnen worden, zoals "()* = , @ '.

< uitdr > of < uitdrukking >

Een uitdrukking kan opgesteld worden door getallen, functies, veldnamen of karakterreeksen op een betekenisvolle manier aan elkaar te voegen. "4 + 8," en "doc = '3' .OR. doc = '4'", zijn uitdrukkingen, net zoals "\$('abc' + &eenreeks,n,3) = 'abcdefg'".

< lijst uitdr >

Een lijst met uitdrukkingen, gescheiden door komma's; meestal worden eenvoudige uitdrukkingen gebruikt. Twee voorbeelden uit de vorige paragraaf zijn vrij ingewikkeld, het eerste voorbeeld daarentegen eenvoudig.

< veld >

De naam van een willekeurig recordveld; in één van de voorbeelden uit de volgende commando's heeft één van de databases veldnamen als ARTIKEL, PRIJS, DATUM, enz.

< veldlijst > of < lijst >

Een lijst met namen van recordvelden, gescheiden door komma's.

< bestand > of < bestandsnaam > of < file >

Een willekeurige bestandsnaam; dit zijn bestandsnamen die moeten voldoen aan de regels voor bestandsnamen, die zijn vermeld in hoofdstuk 3.0.

< formatbestand >

De naam van een rapport-bestandsnaam; zie hoofdstuk 3.4 en het REPORT-commando voor meer informatie over dit soort bestand.

< indexbestand >

De naam van het bestand waarin index-informatie staat; zie hoofdstuk 3.6 en het commando INDEX voor meer informatie over dit soort bestand.

< sleutel >

De veldnaam die voor het indexeren gebruikt wordt; sleutels zijn belangrijk. Een bepaalde database kan verschillende indexen hebben, elk op basis van andere sleutels of een andere combinatie van sleutels. Sleutels kunnen uitdrukkingen zijn of veldnamen. Zie het commando INDEX voor meer informatie.

< geheugenvar >

Een geheugenvariabele; geheugenvariabelen zijn de variabelen die worden gecreëerd door STORE of een commando waarmee een bepaalde waarde wordt bewaard voor later gebruik (ACCEPT, INPUT, enz.). dBASE accepteert maximaal 64 geheugenvariabelen.

< lijst geheugenvariabelen >

Een lijst van door komma's gescheiden geheugenvariabelen.

< n >

Een constante; constanten ("literals") zijn getallen die niet uit geheugenvariabelen of berekeningen verkregen worden. "4 + 8" is geen constante; "4" en "9876" wel.

< bereik >

Een specificatie van het bereik van een commando; het bereik is het gebied dat een commando bestrijkt. Het < bereik > kan drie waarden aannemen:

ALL Alle records in een bestand. ALL betekent dat het bestand wordt teruggespoeld en dat alle records nagekeken worden. ALL is voor bepaalde commando's de standaardwaarde. Voor andere commando's is de standaardwaarde het huidige record (met name voor destructieve commando's als DELETE). Bij elk commando wordt vermeld wat het standaardbereik is. Als u in een willekeurig commando een FOR-uitdrukking gebruikt, zal ALL het standaardbereik zijn.

28 ... REGELS

NEXT n De volgende n records, met inbegrip van het huidige record; NEXT begint met het record waarop de pointer op een bepaald moment staat. De n moet een constante zijn, d.w.z. n mag geen geheugenvariabele of een uitdrukking zijn.

RECORD n Alleen record n; ook in dit geval mag n geen geheugenvariabele of een uitdrukking zijn – n moet een constante zijn.

FOR < uitdr >

Elk record, zolang de logische uitdrukking waar is. Tenzij anders bepaald, worden door een FOR-bepaling ALLE records onderzocht (door het terugspoelen van de database). Deze bepaling kan niet samen met de WHILE-bepaling voorkomen.

WHILE < uitdr >

Werkt op het volgende blok records, zolang een bepaalde logische uitdrukking (< uitdr >) waar is. De uitdrukking moet waar zijn voor het huidige record, anders wordt het commando helemaal niet uitgevoerd. Het commando stopt zodra het systeem de eerste niet-ware uitdrukking tegenkomt. De aanwezigheid van een WHILE-bepaling impliceert NEXT 65534, tenzij anders bepaald; de database wordt niet teruggespoeld. Deze bepaling kan niet samen met de FOR-bepaling voorkomen.

In de commandoformats worden nog andere speciale symbolen gebruikt. Deze symbolen zijn voor elke commando weer anders en worden bij het commando zelf toegelicht.

9.2 GULDEN REGELS

Om het programma goed te laten verlopen, moet voor alle commando-talen een aantal regels nageleefd worden. Onderstaande regels bepalen hoe u uit het algemeen format van een commando de specifieke vorm voor uw toepassing kunt afleiden.

1. Het werkwoord van een commando moet de eerste niet-spatie van de commandoregel zijn; de woorden daarna mogen in een willekeurige volgorde staan. Een werkwoord wijst op een handeling: CREATE, APPEND, REPORT, SET, DISPLAY, en ERASE zijn voorbeelden van werkwoorden - ze hebben een bepaalde handeling tot gevolg. FOR, NEXT en WITH zijn voorbeelden van woorden die het begin vormen van een zinsdeel. Naar al deze woorden wordt verwezen met de term "sleutelwoorden".
2. Om woorden en uitdrukkingen van elkaar te scheiden kunt u een willekeurig aantal spaties gebruiken. Vergeet echter niet dat de spaties ook worden meegeteld bij het berekenen van de limiet van 254 die in Regel 3 beschreven wordt.

3. Alle commando's moeten korter zijn dan 254 tekens (zelfs na een zgn. macro-uitbreiding).
4. Commando's en sleutelwoorden kunnen worden afgekort tot de eerste vier (of meer) tekens. DISPLAY STRUCTURE bijv. kunt u invoeren als DISP STRU of DISPL STRUCT etc. Vergeet echter niet dat de alle letters in een dergelijke afkorting op de juiste manier gespeld moeten worden.
5. Om commando's, sleutelwoorden, veldnamen, geheugenvariabelen of bestandsnamen in te voeren, kunt u hoofdletters of kleine letters gebruiken.
6. Delen van een commando kunnen optioneel zijn, m.a.w. bepaalde delen van een commando kunt u weglaten als het commando gegeven wordt. In het commando-format worden rechte haakjes ([]) gebruikt om aan te duiden welke uitdrukkingen optioneel zijn, en dus weggelaten mogen worden. Dit zijn de uitdrukkingen die gebruikt worden om de werking van de commando's te veranderen. De woorden in hoofdletters zijn de sleutelwoorden en moeten worden ingevoerd als de uitdrukking gebruikt wordt waarin ze voorkomen.
7. Een "gereserveerd woord" is een sleutelwoord dat een fout zal opleveren als het voor iets anders wordt gebruikt dan waarvoor het bedoeld is. In dBASE zijn er geen gereserveerde woorden. Bepaalde veldnamen en bestandsnamen kunnen echter wèl problemen veroorzaken. Voorbeeld: een commandobestand met als naam WHILE zal foutief worden geïnterpreteerd als een DO WHILE-opdracht door de DO-commando-processor; ALL kan in een aantal commando's niet als veldnaam gebruikt worden (ook STATUS, STRUCTURE, FILE, RECORD niet). U kunt dus het gebruik van dBASE-sleutelwoorden als veld- of bestandsnaam beter vermijden.
8. dBase-opdrachten moeten in een commandobestand op de juiste manier worden genest. Iets nesten betekent dat één opdracht in een andere moet passen. Dit is met name belangrijk voor een goede uitvoering van IF-ELSE-ENDIF en DO WHILE-ENDDO. Door in een commandobestand in te springen (indent) kunt u controleren of de opdrachten goed zijn genest. dBASE bemerkt geen nestingsfouten; het commandobestand zal echter op een onvoorspelbare manier worden uitgevoerd. Hieronder volgen een paar voorbeelden voor het goed nesten van deze twee opdrachten.

30 ... REGELS

DO WHILE .NOT. EOF

.
opdrachten

.
IF A .AND. B

.
meer opdrachten

.
ELSE

.
DO WHILE A <= 57

.
nog meer opdrachten

.
ENDDO

.
en nog meer opdrachten

.
ENDIF

.
nog een eindeloos aantal opdrachten

.
ENDDO

Zo wordt op de juiste manier genest. De IF-ELSE-ENDIF-opdracht zit volledig in de DO WHILE-ENDO-opdracht, net zoals de tweede DO WHILE-ENDO volledig in het ELSE-gedeelte van de IF-ELSE-ENDIF zit. We zouden nog meer nestingsniveaus kunnen laten zien, omdat dBASE veel meer niveaus accepteert.

DO WHILE .NOT. EOF

.
opdrachten

.
IF iets verandert in waarde

.
ENDDO

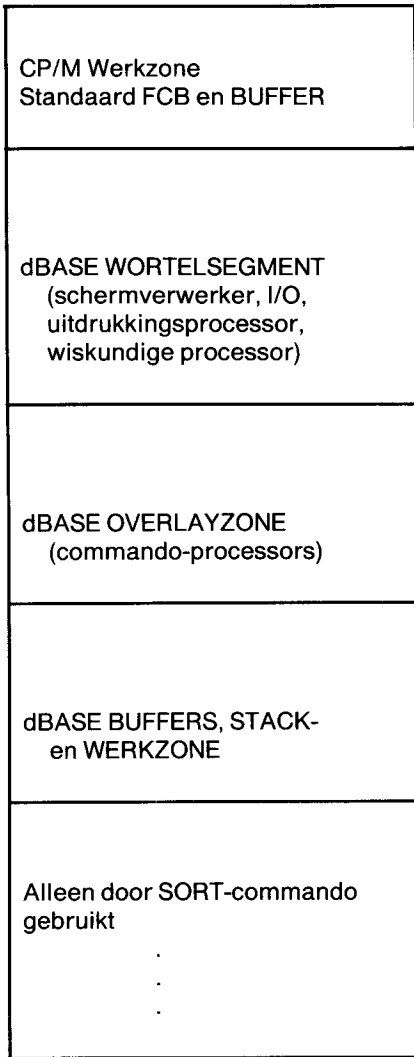
.
andere opdrachten

.
ENDIF

Dit is FOUT. De ENDDO gaat over de grenzen van de IF-ENDIF-groep; de twee opdrachten zijn dus niet goed genest. Het commandobestand waarin deze opdrachten staan zal niet verlopen zoals verwacht en dBASE zal hier geen verklaring voor geven.

10.0 MACHINETAAL-INTERFACE

10.1 GEHEUGENKAART



dBASE
A 400 2.3
B 000 2.4

BDOS tot geheugentop
(CCP in overlay)

32 ... MACHINETAAL

10.2 MACHINETAAL-COMMANDO'S

SET CALL TO < adres >

Stelt het decimaal adres in dat door het dBASE CALL-commando zal worden aangeroepen.

CALL [< geheugenvar >]

Voert een machinetaal-aanroep uit naar het adres dat ingesteld is met SET CALL TO of naar het standaardadres als geen SET CALL gedaan werd. Er is een stack van ongeveer 254 bytes beschikbaar, het HL-registerpaar wijst naar de eerste byte als < geheugenvar > een tekenreeks is. U mag niet proberen een tekenreeks langer of korter te maken. Met de instructie RET keert u terug naar dBASE besturing.

LOAD [< bestand >]

Dit commando slaat een bestand (.HEX-bestand met INTEL HEX-format) op in het geheugen.

POKE < adres >, < gegevensbyte > [, < gegevensbyte > ...]

Plaatst de gegevens rechtstreeks in het geheugen.

PEEK (< adres >)

Een functie waarmee het getal verkregen wordt dat overeenkomt met de (positieve) binaire waarde van de byte op < adres >.

Commando's

?

```
?[ < lijst uitdrukkingen > ]
??[ < lijst uitdrukkingen > ]
```

Dit commando is een speciale vorm van het DISPLAY-commando en heeft dezelfde werking als DISPLAY OFF < uitdr > . Het kan gebruikt worden om de waarde van een uitdrukking of een reeks uitdrukkingen weer te geven. Het vraagtekencommando (als "wat is" uitgesproken) kan zowel geheugenvariabelen, databasevelden, constanten of functies weergeven. Een "?" zonder uitdrukking geeft een regelsprong: in een commandobestand kunt u hiermee de informatie op het scherm "spreiden".

De tweede vorm van dit commando "??" heeft dezelfde functie als "?", maar print de uitdrukking zonder voorafgaande regelsprong. Dit commando kan in commandobestanden worden gebruikt om meer dan één uitdrukking op dezelfde regel weer te geven.

Als SET RAW ON in werking is worden geen spaties tussen de items van een lijst geplaatst; is dat niet het geval, dan worden de items door één spatie van elkaar gescheiden.

Voorbeelden

```
.USE TEL__BSTD
.4
.? #
4
.? NAAM
BROUWERS, JAN
.? 5+9
14
```

Hieronder een voorbeeld van een commandobestand waarin ? gebruikt wordt om de informatie te spreiden. Het commandobestand is opgesteld om met het commando "DBASE H:FILE" uitgevoerd te worden; op de volgende pagina ziet u wat er in zo'n geval gebeurt.

SET DEFAULT TO G
USE trace INDEX trace
DISP STRU
?
ACCEPT "Voer datum in" TO dte
SET DATE TO &dte
RELEASE dte
RETURN

STRUCTURE FOR FILE: TRACE.DBF
NUMBER OF RECORDS: 02359
DATE OF LAST UPDATE: 10/08/82
PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
001	UP	C	024	
002	TRFLD	C	005	
003	DOC	C	024	
004	DESCR	C	080	
005	NATURE	C	010	
006	STATUS	C	006	
007	TESTED	C	004	
TOTAL			00154	

Voer datum in: 14/10/81

@

@ <coördinaten> [SAY <uitdr> [USING <format>]]
 [GET <variabele> [PICTURE <format>]]

Dit commando werkt met de commando's SET FORMAT TO, ERASE, EJECT, CLEAR GETS en READ en is een efficiënte manier om informatie geformatteerd op het scherm of op de printer weer te geven. De manier waarop een "@" ("at-sign", ook wel "apestaartje") geïnterpreteerd wordt, hangt af van de SET FORMAT TO-status, die ook bepaalt of de andere commando's al dan niet werken. Alle combinaties worden hierna besproken.

De <coördinaten> zijn een "x,y"-paar en kunnen twee betekenissen hebben: scherm- of printercoördinaten. De "x,y" duidt een regel (x) en een kolom (y) aan. Op de meeste beeldschermen hebben de schermcoördinaten een "x"-bereik van 0-23 en een "y"-bereik van 0-79, m.a.w. 24 regels en 80 kolommen. Op de 0-regel geeft dBASE meldingen aan de gebruiker: daarom laat u deze regel bij voorkeur vrij. De printer-coördinaten hebben een "x"- en een "y"-bereik van 0-254. In beide gevallen kunnen de coördinaten constanten, numerieke waarden, geheugenvariabelen of numerieke uitdrukkingen zijn. Het commando SET FORMAT TO bepaalt of de coördinaten als scherm- of als printer-coördinaten geïnterpreteerd worden.

De <coördinaten> mogen ook relatieve adressen zijn. In dat geval refereren ze naar de laatst gebruikte scherm- of printerpositie. Een constructie "\$ + <uitdrukking>" bepaalt het relatief adres als volgt: de <coördinaat> waar het laatste @-commando ophield plus de waarde van de uitdrukking. Als op een @ 10,10 say 'HALLO' het commando @ \$,\$+6 say 'DAAR' zou volgen, krijgen we op regel 10 kolom 10 de tekst 'HALLO DAAR'. Zowel rij als kolom mogen relatief geadresseerd worden. Alleen positieve verschuivingen zijn geldig (b.v. \$ - <uitdrukking> is NIET toegestaan).

Bij SET FORMAT TO SCREEN (standaard) brengt het "@"-commando de gegevens naar het scherm. Een coördinatenpaar 0,0 verwijst naar de eerste tekenpositie in de linkerbovenhoek van het scherm (ook wel start- of home-positie genoemd). Het paar 10,15 verwijst naar regel 11 en kolom 16 van het scherm. Nogmaals, regel 0 bij voorkeur niet gebruiken. "@"-commando's mogen in elke volgorde naar het scherm gestuurd worden. M.a.w. men mag iets afdrukken op regel 15 en daarna iets afdrukken op regel 10. Hetzelfde geldt voor de kolommen.

Bij SET FORMAT TO PRINT drukt het "@"-commando de gegevens op de printer. Het coördinatenpaar 0,0 verwijst naar de linker bovenhoek van het papier. "@"-commando's naar de printer moeten wél in de juiste volgorde worden gegeven, omdat er anders heel wat papier verspild wordt. Ga ervan uit dat u de gegevens op een schrijfmachine typt: commando's voor regel 5 mogen niet ná commando's voor regel 6 komen. Commando's voor kolom 20 moeten volgen op commando's voor kolom 10. Als de printer een commando ontvangt dat een "terugkeer" impliceert, zal hij een paginasprong maken en het vervolg van de informatie op een nieuwe pagina printen.

Bij SET FORMAT TO SCREEN gebruikt u het commando ERASE om het scherm te wissen: alle voorgaande informatie wordt gewist, alle GETS komen vrij (zie verder) en de coördinaten worden op 0,0 gesteld. Bij SET FORMAT TO PRINT gebruikt u EJECT om naar een nieuwe pagina te springen en de coördinaten weer op 0,0 in te stellen.

Een SAY-zin dient om een uitdrukking op het scherm te brengen welke later niet door het READ commando gewijzigd kan worden. De USING-sub-zin formatteert de uitdrukking in de SAY-zin. Formateer-instructies worden hieronder behandeld. Het is raadzaam steeds de USING-subzin te gebruiken, omdat dBASE de uitdrukking zonder USING mogelijk op een ongewenste manier interpreteert.

SAY-zinnen kunnen zowel met het scherm als met de printer worden gebruikt. GET's worden echter alleen herkend bij SET FORMAT TO SCREEN.

De GET-zin geeft de huidige waarde van een veldvariabele of geheugenvariabele weer. De variabele moet bestaan voor de GET gegeven wordt en kan later met het READ-commando gewijzigd worden. De PICTURE-zin kan worden gebruikt met een GET-zin voor speciale formattering en om te kijken of de ingevoerde gegevens geldig zijn (zie het READ-commando voor meer informatie). Zonder PICTURE-bepaling fungeert het gegevenstype (teken, numeriek of logisch) als impliciete PICTURE.

Als het gegevenstype van de veld- of geheugenvariabele in de GET een logische waarde is, dan accepteert de tekencontrole alleen de tekens 'T', 'F', 'Y', 'N' (voor True/False en Yes/No) in hoofdletters of in kleine letters.

Er mogen maximaal 64 GET's tegelijk actief zijn. Om de bestaande GET's vrij te maken gebruikt u de commando's ERASE of CLEAR GETS.

Bij SET FORMAT TO SCREEN vult een zonder SAY- of GET-zin de rest van de regel, die door de coördinaten aangegeven is met spaties. @ 10,0 zal regel 11 dus helemaal wissen.

Bij SET FORMAT TO SCREEN moet er een READ worden gegeven om de GET's in te vullen (zie READ-commando). Bij SET FORMAT TO PRINT hoeft op de "@"-commando's geen READ-commando te volgen.

Aangezien er geen READ nodig is om te printen, kan de gebruiker de output op voorbedrukte formulieren (b.v. cheques, bestelbonnen, enz.) moeiteloos direct formatteren: onthoud alleen dat u de "@"-commando's 'sequentieel' moet geven, net als op een schrijfmachine.

SET FORMAT TO PRINT dient vaak om meer dan één item te printen. De mogelijkheid om als coördinaten geheugenvariabelen te gebruiken komt hier goed van pas. Het volgende voorbeeld laat zien hoe u vanuit een commando-bestand een speciaal rapportformulier kunt opmaken.

```

SET FORMAT TO PRINT
GOTO TOP
STORE 7 TO TELLER
DO WHILE .NOT. EOF
  IF TELLER > = 50
    EJECT
    STORE 10 TO TELLER
  ENDIF
  @ TELLER,12 SAY P USING 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
  @ TELLER,48 SAY D USING 'XXXXXXXXXX'
  @ TELLER,64 SAY P1 USING 'XXXXXXXXXXXXXXXXXXXX'
  @ TELLER,88 SAY U USING 'XXXXXXXXXX'
  @ TELLER,104 SAY P2 USING 'XXXXXXXXXXXXXXXXXXXX'
  IF RCD < > 0
    @ TELLER,130 SAY RCD USING '9999'
  ENDIF
  STORE TELLER + 1 TO TELLER
  SKIP
ENDDO
RETURN

```

Dit commando-bestand drukt op elke regel de inhoud van een aantal velden (één veld wordt overigens selectief geprint) uit records van een database. De eerste regel van elke pagina wordt op regel 10 geprint. Na 50 print-regels wordt een nieuwe pagina gestart (EJECT). De SAY-zin moet genoeg ruimte reserveren voor de variabele. Als de USING korter is dan de variabele of het veld, dan worden deze afgebroken. Het <format> voor USING-tekenreeksen wordt in de tabel hierna toegelicht.

Als in de SET FORMAT TO PRINT-modus de coördinaten van de volgende “@” bepalen dat informatie op dezelfde regel geprint wordt maar in een reeds gebruikte kolom, dan kan de printer zich onvoorspelbaar gaan gedragen. Meestal zal de printkop helemaal naar rechts springen en daar alle gegevens op één enkele positie over elkaar heen printen. Bij SET FORMAT TO SCREEN wordt de nieuwe informatie over de oude heen geschreven.

De laatste vorm van het SET FORMAT commando is SET FORMAT TO <formatbestand>. Als deze modus gekozen is worden na een READ-commando de “@”-commando’s gelezen (READ) uit het vooraf ontworpen <formatbestand>. Hiermee kan de gebruiker formats (scherm-indelingen) opmaken voor bijzondere toepassingen. Onthoud evenwel dat formatbestanden geen vereiste zijn om het “@”-commando te kunnen gebruiken aangezien “@”’s ook in de commando-bestanden zelf mogen staan. Zie READ voor meer informatie.

Formats:

Zowel USING als PICTURE dienen om een format te bepalen. Het format is een reeks symbolen die aangeven welke karakters op het scherm of op het blad moeten verschijnen. In de volgende lijst worden deze symbolen en hun functie gedefinieerd:

Formatteken	SAY-functie	GET-functie
#	volgende getal wordt afgedrukt (output)	alleen een cijfer (1,2,...,8, 9, 0) en de tekens . , + - en spatie mogen worden ingevoerd
9	als bij #	als bij #
X	geeft volgend teken	elk teken mag worden ingevoerd
A	geeft volgend teken	alleen letters mogen worden ingevoerd
\$ of *	geeft een cijfer of \$ of een * i.p.v. voorafgaande nullen	wordt letterlijk overgenomen
!	geen werking	zet kleine letters om hoofdletters

Voorbeeld:

```
.@ 5,1 SAY 'VOER TELEFOONNUMMER IN' GET TELNR PICTURE
'(999)999-9999'
```

Als we ervan uitgaan dat de waarde van TELNR aanvankelijk helemaal blanco was, dan verschijnt op het scherm de prompt 'VOER TELEFOONNUMMER IN', gevolgd door '(sss)sss-sss' (s staat voor spatie). Wanneer een READ-commando wordt gegeven kunnen nu alleen cijfers worden ingevoerd. De waarde van TELNR na het READ-commando zou na het wijzigen b.v. '(213)555-5555' kunnen zijn. Alle niet-functionele tekens in het PICTURE-format worden in de variabele letterlijk overgenomen. In dit voorbeeld zijn dat de haakjes, het min-teken en de spatie.

```
.@ 10,50 SAY UREN*LOON USING '*****.99'
```

Dit "@"-commando kan zowel op het scherm als op de printer gebruikt worden aangezien er geen GET-zin in staat. Dit commando zou b.v. kunnen dienen om looncheques te printen. De nullen die aan het begin van het te printen item staan worden door sterretjes vervangen. Als uren=40 en loon=12.50, dan wordt '***500.00' weergegeven. Door de sterretjes is zo'n bedrag achteraf moeilijk te vervalsen.

Wanneer in het gedeelte voor de (decimale) punt van een PICTURE komma's (voor de duizendtallen) voorkomen verschijnen die alleen wanneer ze voorafgegaan worden door significante cijfers; in elk ander geval verschijnt het teken dat normaliter voorafgaande nullen vervangt. Voorbeeld:

```
.@ 10.50 SAY UREN*LOON USING '***,***.99'
```

Dit zal ****500.00 en niet ***,500.00 als resultaat hebben.

Gewoonlijk wordt een reeks "@"-commando's gegeven en indien deze GET-zinnen bevatten, volgt een READ-commando om wijzigingen en invoer van gegevens in de GET-variabelen mogelijk te maken. In het volgende voorbeeld is het scherm geformatteerd met behulp van een aantal "@"-commando's waarmee informatie naar een database geschreven. Het laatste record heeft een "0" in het veld "naam"; dit zal gewist worden omdat het overbodig is.

```
SET FORMAT TO SCREEN
USE F:TEL_BSTD
DO WHILE NAAM # '0'
  APPEND BLANK
  @ 5,0 SAY "VOER VOLGENDE NAAM IN";
  GET NAAM PICTURE 'XXXXXXXXXXXXXXXXXXXX'
  @ 6,0 SAY "VOER TOESTELNUMMER IN";
  GET TST:NR PICTURE 'XXXXX'
  @ 6,40 SAY "VOER AFDELINGNUMMER IN";
  GET AFDELING PICTURE 'XXXXXXXXXXXX'

  READ
ENDDO
GOTO BOTTOM
DELETE
PACK
LIST
RETURN
```

De volgende commando's beïnvloeden de werking van het "@"-commando:

- SET INTENSITY ON/OFF (ON is standaard) bepaalt de helderheid bij GET's en SAY's.
- SET BELL ON/OFF (ON is standaard) stelt de biepton in of uit bij invoering van ongeldige tekens of te veel gegevens.
- SET COLON ON/OFF (ON is standaard) bepaalt of GET-variabelen al dan niet met dubbele punten worden afgebakend.
- SET DEBUG ON/OFF (OFF is standaard) om fouten in "@"-commando's op te sporen door ECHO- en STEP-berichten naar de printer te sturen.
- SET SCREEN ON/OFF (ON is standaard) voor full-screen werking.
- SET FORMAT TO SCREEN/PRINT/ <formatbestand> bepaalt de uitvoer-eenheid (SCHERM of PRINTER). SET FORMAT TO <formatbestand> kiest een formatbestand voor "@"-commando's voor het READ-commando. SCREEN (scherm) is standaard.
- READ schakelt de EDIT-modus in, zodat GET-variabelen gewijzigd kunnen worden.

ACCEPT

ACCEPT ["<tekenreeks>"] TO <geheugenvar>

Met deze constructie kunnen net als met het INPUT-commando tekenreeksen in geheugenvariabelen worden opgenomen. In tegenstelling tot INPUT vereist ACCEPT geen aanhalingstekens: ACCEPT zet elke invoer om in een geheugenvariabele van het type 'teken'. INPUT bepaalt daarentegen het gegevenstype op basis van de syntaxis van de invoer en maakt vervolgens een geheugenvariabele van het gepaste type.

De <geheugenvar> wordt – indien nodig – gecreëerd en de ingevoerde tekenreeks wordt in <geheugenvar> opgeslagen. Als een "<tekenreeks>" gegeven is verschijnt deze als prompt op het scherm, gevolgd door een dubbele punt. Als het antwoord op het ACCEPT-verzoek een return is, dan krijgt <geheugenvar> één spatie als waarde. Om de promptreeks aan te geven mogen dubbele aanhalingstekens, enkele aanhalingstekens of rechte haakjes gebruikt worden; aan begin en einde ze moeten echter wel van dezelfde soort zijn.

Voorbeelden:

```
.ACCEPT "VOER NAAM IN" TO NAM
VOER NAAM IN:Anna Bakker
```

```
.ACCEPT "VOER NAAM IN" TO NAM2
VOER NAAM IN:Paul Smits
```

```
.DISP MEMO
NAM          (C)  Anna Bakker
NAM2         (C)  Paul Smits
**TOTAL**           02 VARIABLES USED           00021 BYTES USED
```

```
.ACCEPT TO ZOMAARWAT
:WILLEKEURIGE TEKENS
```

```
.DISP MEMO
NAM          Anna Bakker
NAM2         Paul Smits
ZOMAARWAT    WILLEKEURIGE TEKENS
**TOTAL**           03 VARIABLES USED           00043 BYTES USED
```

APPEND

- a. APPEND FROM <bestand> [FOR <uitdr>] [SDF] [DELIMITED]
[WHILE <uitdr>]
- b. APPEND BLANK
- c. APPEND

Met deze drie vormen worden records toegevoegd aan de database in USE. APPEND, CREATE en INSERT zijn de enige commando's die records aan een bestand toevoegen. Met APPEND en CREATE kunnen verschillende records achter elkaar ingevoerd worden; INSERT voegt slechts één record toe.

De eerste vorm voegt records toe vanuit een ander <bestand>. Als de SDF-bepaling gegeven is, dan wordt verondersteld dat de records in het System Data Format staan (zie hoofdstuk 6.0). Als de nieuwe records kleiner zijn dan de oude records in het USE-bestand, dan worden de nieuwe records rechts opgevuld met spaties; als de nieuwe records langer zijn dan de records uit het USE-bestand, dan worden de toegevoegde records afgebroken. Records worden aan het USE-bestand toegevoegd totdat het systeem in het FROM-bestand een end-of-file (einde bestand) aantreft.

Als DELIMITED in het APPEND-commando voorkomt, neemt dBASE aan dat de records in het FROM-bestand scheidingstekens hebben en verwerkt deze dan op de gepaste manier. Veel computertalen creëren bestanden waarbij tekenreeksen tussen enkele of dubbele aanhalingstekens staan en velden door komma's worden gescheiden. In de DELIMITED-modus verwijdert dBASE deze tekens uit zulke bestanden en worden de gegevens opgenomen in een database met dBASE-structuur (volgens de structuur van die database).

Als noch SDF noch DELIMITED vermeld worden gaat dBASE ervan uit dat het FROM-bestand een database-bestand is met dBASE-structuur. De structuren van het USE- en FROM-bestand worden met elkaar vergeleken. Velden die in de records van beide bestanden voorkomen, worden uit het FROM-bestand gelezen en aan het USE-bestand toegevoegd. Opvullen en afbreken gebeurt waar nodig zo dat de gegevensitems uit het FROM-bestand in de structuur van het USE-bestand passen.

Als de FOR-zin gebruikt wordt voegt dBASE de records in het FROM-bestand één voor één toe, waarbij de FOR-voorwaarde telkens getest wordt. Dit verloopt als volgt: eerst voegt dBASE een record toe en daarna test hij de voorwaarde. Als de uitdrukking waar is wordt het record behouden, terwijl als de uitdrukking niet-waar is het record genegeerd wordt. Deze procedure gaat door tot de end-of-file van het FROM-bestand. Aangezien dBASE het record eerst toevoegt, is het duidelijk dat de uitdrukking in de FOR-zin alleen velden mag testen van het bestand dat de records *ontvangt*.

Als de BLANK-bepaling (vorm b.) gegeven is, dan wordt aan het USE-bestand één record toegevoegd dat opgevuld is met spaties. Dit record kan dan met de EDIT of REPLACE-opdrachten ingevuld worden.

42 ... APPEND

Als op APPEND geen bepalingen volgen (vorm c.) dan krijgt de gebruiker een prompt met de veldnamen uit de structuur van het USE-bestand. Vanaf het toetsenbord kan een willekeurig aantal records toegevoegd worden. Een < return > op de eerste positie van het eerste veld van een record beëindigt APPEND.

Als de database in USE geïndexeerd is, worden de indexbestanden die in het USE-commando gegeven werden, automatisch bijgewerkt wanneer de nieuwe records worden toegevoegd (behalve voor APPEND BLANKs). Elk ander indexbestand voor die database moet opnieuw geïndexeerd worden.

Als APPEND in de full-screen-modus gebruikt wordt, dan brengt SET CARRY ON alle gegevens uit het voorgaande record over naar het volgende, waarna vervolgens wijzigingen kunnen worden aangebracht. Dit is vooral interessant als in opeenvolgende records veel gegevens identiek zijn.

Als een SET FORMAT TO <bestand> in werking is, dan gebruikt APPEND de "@"-commando's uit het formaatbestand om het scherm in te delen en gegevens in de gewenste vorm te laten toevoegen. Anders geeft APPEND alle velden in tabelvorm weer met de namen die zij bij de structuurbepaling kregen.

Het APPEND-commando is zeer interessant wanneer velden uitgebreid of ingekort moeten worden of om velden aan een bestaande database toe te voegen of eruit te wissen. Met het CREATE-commando wordt dan eerst een nieuwe database opgebouwd met de gewenste structuur waarna met APPEND de oude database aan de nieuwe wordt toegevoegd. Velden die alleen in de nieuwe database voorkomen, zullen met spaties gevuld worden.

Voorbeelden:

```
.USE TEL__BSTD
```

```
.DISPLAY STRUCTURE
```

```
STRUCTURE FOR FILE:      TEL__BSTD
```

```
NUMBER OF RECORDS:      00005
```

```
DATE OF LAST UPDATE:    31/12/82
```

```
PRIMARY USE DATABASE
```

FLD	NAME	TYPE	WIDTH	DEC
001	NAAM	C	020	
002	TST:NR	C	005	
003	AFDELING	C	010	
TOTAL			00036	

```
.DISPLAY ALL
```

00001	TIMMERMANS, THEO	1357	123/456
00002	KNOOPS, HERMAN	2468	180/103
00003	BERINGS, MARIJKE	3344	264/401
00004	SMULDERS, KEES	6743	190/901
00005	PAPAMOROULOS, D	1011	84/138

.APPEND

RECORD 00006

NAAM: DE RUITER, JAN-JORIS
 TST:NR: 6623
 AFDELING: 170/430

RECORD 00007

NAAM: DE JONG, ELLEN
 TST:NR: 8093
 AFDELING: 427/396

RECORD 00008

NAAM: < return >

.DISPLAY ALL OFF NAAM, TST:NR

TIMMERMANS, THEO	1357
KNOOPS, HERMAN	2468
BERINGS, MARIJKE	3344
SMULDERS, KEES	6743
PAPAMOROULOS, D	1011
DE RUITER, JAN-JORIS	6623
DE JONG, ELLEN	8093

.APPEND FROM DUPE3
 00007 RECORDS ADDED

.DISPLAY ALL

00001	TIMMERMANS, THEO	1357	123/456
00002	KNOOPS, HERMAN	2468	180/103
00003	BERINGS, MARIJKE	3344	264/401
00004	SMULDERS, KEES	6743	190/901
00005	PAPAMOROULOS, D	1011	84/138
00006	DE RUITER, JAN-JORIS	6623	170/430
00007	DE JONG, ELLEN	8093	427/396
00008	TIMMERMANS, THEO	1357	
00009	KNOOPS, HERMAN	2468	
00010	BERINGS, MARIJKE	3344	
00011	SMULDERS, KEES	6743	
00012	PAPAMOROULOS, D	1011	
00013	DE RUITER, JAN-JORIS	6623	
00014	DE JONG, ELLEN	8093	

44 ... APPEND

.APPEND BLANK

.DISPLAY
00015

.REPLACE NAME WITH "VAN DEUTEKOM, MARIEKE"

00001 REPLACEMENT(S)

.DISPLAY

00015 VAN DEUTEKOM, MARIEKE

.DISPLAY ALL NAAM, 'toestel =',

TST:NR

00001	TIMMERMANS, THEO	toestel =	1357
00002	KNOOPS, HERMAN	toestel =	2468
00003	BERINGS, MARIJKE	toestel =	3344
00004	SMULDERS, KEES	toestel =	6743
00005	PAPAMOROULOS, D	toestel =	1011
00006	DE RUITER, JAN-JORIS	toestel =	6623
00007	DE JONG, ELLEN	toestel =	8093
00008	TIMMERMANS, THEO	toestel =	1357
00009	KNOOPS, HERMAN	toestel =	2468
00010	BERINGS, MARIJKE	toestel =	3344
00011	SMULDERS, KEES	toestel =	6743
00012	PAPAMOROULOS, D	toestel =	1011
00013	DE RUITER, JAN-JORIS	toestel =	6623
00014	DE JONG, ELLEN	toestel =	8093
00015	VAN DEUTEKOM, MARIEKE	toestel =	

.USE B:INKOPEN

.DISP STRU

STRUCTURE FOR FILE: B:INKOPEN.DBF

NUMBER OF RECORDS: 00009

DATE OF LAST UPDATE: 22/06/82

PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
001	ITEM	C	020	
002	AANTAL	N	005	
003	PRIJS	N	010	002
TOTAL			00036	

.CREATE

FILENAME: NIEUWINK

ENTER RECORD STRUCTURE AS FOLLOWS:

FIELD NAME, TYPE, WIDTH, DECIMAL PLACES

001 ITEM, C, 25

002 AANTAL, N, 5

003 PRIJS, N, 10, 2

004 DAT:INK, C, 8

005 < return >

INPUT NOW? N

.USE NIEUWINK

.APPEND FROM B:INKOPEN

00009 RECORDS ADDED

.LIST

00001	BONEN	5	0.75
00002	BROOD	2	0.97
00003	BIEFSTUK	4	3.94
00004	PAPIEREN BORDEN	1	0.86
00005	PLASTIC VORKEN	5	0.42
00006	KROPSLA	2	0.53
00007	KAAS	1	1.96
00008	MELK	2	1.30
00009	HOUTSKOOL	2	0.75

.REPLACE ALL DAT: INK WITH '7/4/82'

00009 REPLACEMENT(S)

00001	BONEN	5	0.75	4/ 7/82
00002	BROOD	2	0.97	4/ 7/82
00003	BIEFSTUK	4	3.94	4/ 7/82
00004	PAPIEREN BORDEN	1	0.86	4/ 7/82
00005	PLASTIC VORKEN	5	0.42	4/ 7/82
00006	KROPSLA	2	0.53	4/ 7/82
00007	KAAS	1	1.96	4/ 7/82
00008	MELK	2	1.30	4/ 7/82
00009	HOUTSKOOL	2	0.75	4/ 7/82

(Hieronder een voorbeeld van een APPEND met een DELIMITED-bestand.
Dit bestand zou met diverse versies van BASIC aangemaakt kunnen zijn.)

'BAARD, KAREL',31415,6
'MEIERS, MIRIAM',76767,17
'ADAMS, LOUISE',92653,15
'BLOEMEN, HUGO',11528,16

(Voeg het bestand toe aan een database met dBASE-structuur)

.USE BESTEL

.DISP STRU

STRUCTURE FOR FILE: BESTEL.DBF

NUMBER OF RECORDS: 00008

DATE OF LAST UPDATE: 00/00/00

PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
001	KLANT	C	020	
002	ART:NR	C	005	
003	AANTAL	N	010	
TOTAL			00036	

46 ... APPEND

```
.LIST
00001 TILLAERTS, TINEKE      31415      13
00002 MEYKOORTS, HOKKE     76767      13
00003 CHANG, LEE           11528      44
00004 ADAMS, LOUISE        89793      12
00005 GOOSSENS, GUUS       31415       3
00006 JOOSTEN, DICK        76767       5
00007 SCHILPEROORT, PIET   21828       5
00008 TARAM, CLARA         70296       9
```

```
.APPEND FROM DELIM.DAT DELIMITED
00004 RECORDS ADDED
```

```
.LIST
00001 TILLAERTS, TINEKE      31415      13
00002 MEYKOORTS, HOKKE     76767      13
00003 CHANG, LEE           11528      44
00004 ADAMS, LOUISE        89793      12
00005 GOOSSENS, GUUS       31415       3
00006 JOOSTEN, DICK        76767       5
00007 SCHILPEROORT, PIET   21828       5
00008 TARAM, CLARA         70296       9
00009 BAARD, KAREL         31415       6
00010 MEIERS, MIRIAM        76767      17
00011 ADAMS, LOUISE        92653      15
00012 BLOEMEN, HUGO        11528      16
```

(Hieronder een voorbeeld van APPEND FROM <bestand> FOR <uitdr>. Zoals u ziet komen de velden in het FOR-bestand ook in het USE-bestand voor.)

```
.USE CHEQUES
.DISP STRU
STRUCTURE FOR FILE:      CHEQUES.DBF
NUMBER OF RECORDS:      00013
DATE OF LAST UPDATE:    19/10/83
PRIMARY USE DATABASE
FLD      NAME           TYPE      WIDTH    DEC
001      GETAL          N         005
002      AAN              C         020
003      BEDRAG          N         010      002
004      INKOMEND         L         001
005      UITGAAND          L         001
**TOTAL**                00038
```

```
.LIST
00001      1  Telefoon          104.89 .F. .T.
00002      2  Gas              4.14 .F. .T.
00003      3  Elektriciteit    250.31 .F. .T.
00004      4  Supermarkt       1034.45 .F. .T.
00005     34  lk              561.77 .T. .F.
00006      6  Bankkosten        4.00 .T. .T.
00007      7  Doctor Peeters   100.00 .T. .T.
00008      8  Omroepvereniging  101.01 .F. .T.
00009      9  Reparatie auto    500.01 .T. .T.
00010     10  lk              561.01 .T. .F.
00011     11  Tupperware        50.02 .F. .T.
00012     12  lk              561.77 .T. .F.
00013     13  lk              750.03 .T. .F.
```

```
.USE MAAND
```

```
.DISP STRU
```

```
STRUCTURE FOR FILE:      MAANC DBF
```

```
NUMBER OF RECORDS:      00003
```

```
DATE OF LAST UPDATE:    18/10/82
```

```
PRIMARY USE DATABASE
```

FLD	NAME	TYPE	WIDTH	DEC
001	GETAL	N	005	
002	AANTAL	N	010	002
003	INKOMEND	L	001	
TOTAL			00017	

```
.LIST
```

```
00001      29              14.89 .T.
```

```
00002      16             764.09 .T.
```

```
00003      78             97.96 .T.
```

```
.APPEND FROM CHEQUES FOR INKOMEND
```

```
00007 RECORDS ADDED
```

```
.APPEND FROM CHEQUES FOR UITGAAND
```

```
*** SYNTAX ERROR ***
```

```
?
```

```
.APPEND FROM CHEQUES FOR UITGAAND
```

```
CORRECT AND RETRY (Y/N)? N
```

Het laatste voorbeeld werd gegeven om te laten zien wat er gebeurt als u in FOR een veld gebruikt dat niet in het USE-bestand aanwezig is.

BROWSE

BROWSE [FIELDS < lijst velden >]

Het BROWSE-commando is een van de krachtigste dBASE-commando's voor het wijzigen en bekijken van gegevens. Gegevens van max. 19 records worden op het scherm weergegeven (minder, als de velden breder zijn dan 80 tekens). Er worden zoveel mogelijk velden op één regel geplaatst. Het scherm moet worden gezien als een venster in de database. U kunt voor- en achteruit door de records rollen en u kunt naar links en rechts door de velden van de database lopen. De gegevens kunnen worden gewijzigd met de normale full-screen edit-toetsen (zie hoofdstuk 8 voor extra informatie).

Als er geen < veldlijst > is bepaald, dan zal BROWSE alle velden in dezelfde volgorde als in de structuur laten zien.

Hieronder een samenvatting van de schermcontrotoetsen die na het BROWSE-commando gebruikt kunnen worden:

- < control > -E,A gaat terug naar het vorige gegevensveld;
- < control > -X,F gaat vooruit naar het volgende gegevensveld;

- < control > -D gaat vooruit naar het volgende teken;
- < control > -S gaat achteruit naar het vorige teken;

- < control > -G wist het teken waarop de cursor staat;
- < rubout > wist het teken voor de cursor;

- < control > -Q verlaat zonder wijzigingen te bewaren;
- < control > -W verlaat en bewaart wijzigingen (< control > -O voor Superbrain);

- < control > -B verschuift het venster één veld naar rechts;
- < control > -Z verschuift het venster één veld naar links;

- < control > -C schrijft het lopend record weg en gaat één record vooruit;
- < control > -R schrijft het lopend record weg en gaat één record achteruit;

- < control > -U verandert de aan- of afwezigheid van een wismarkering van het lopend record.

Voorbeeld:

.BROWSE

.BROWSE FIELDS FIRMANAAM, DIRECTEUR, TELEFOON

CANCEL

CANCEL

Annuleert de uitvoering van een commandobestand en keert terug naar normale besturing via het toetsenbord;

Voorbeeld:

```
INPUT 'IS JOB KLAAR (Y = JA, N = NEE)' TO X
IF X
  CANCEL
ENDIF
```

Dit is een gedeelte uit een commandobestand. Het INPUT-commando vraagt om een YES/NO antwoord. Als het antwoord YES is (Y, y, T of t) wordt aan de IF X-regel van het commando-bestand voldaan (omdat X logisch .TRUE. is) en wordt het CANCEL-commando uitgevoerd.

50 ... CHANGE

CHANGE

CHANGE [< bereik >] FIELD < lijst > [FOR < uitdr >]

CHANGE is een commando waarmee de gebruiker zonder veel moeite een aantal wijzigingen kan aanbrengen in een database. Alle velden uit de database die in de lijst worden opgesomd, worden de gebruiker in de volgorde gegeven die in de < lijst > bepaald is. De gebruiker heeft de mogelijkheid om nieuwe gegevens in te voeren, gegevens te wijzigen of over te springen naar het volgende veld. Als er niets meer in de < lijst > staat, gaat CHANGE over naar het volgende record, dat in < bereik > bepaald is. Het standaard-bereik is het huidige record.

Een veld kan volledig worden gewist door < control > -Y in te toetsen (gevolgd door < return >) als antwoord op de boodschap CHANGE?. Het CHANGE-commando kan worden afgebroken door een < escape > aan te slaan.

Voorbeeld:

```
.USE VOORRAAD  
.CHANGE FIELD DATUM
```

```
RECORD: 00001
```

```
DATUM: 08/10/83  
CHANGE? 83  
TO          84
```

```
DATUM: 08/10/84  
CHANGE? < return >
```


CLEAR**CLEAR [GETS]**

Met het GETS (of GET) sleutelwoord maakt CLEAR alle lopende GET's (d.w.z. elke GET ingesteld door een @-commando) vrij en blijft het scherm onveranderd. Dit in tegenstelling tot het ERASE-commando, dat niet alleen de lopende GET's maar ook het hele scherm wist.

Zonder het GETS-sleutelwoord zorgt CLEAR voor een volledige 'reset' van dBASE. Alle databases in USE worden gesloten en komen in "un-used" status, alle geheugenvariabelen worden vrijgemaakt en de PRIMARY-werkzone wordt opnieuw geselecteerd.

Na dit commando begint dBASE met een "schone lei". Voorbeeld: als een commando-bestand is uitgevoerd en dBASE in de SECONDARY-status is gebleven, dan kunnen er onvoorspelbare dingen gebeuren als een nieuw commando-bestand wordt uitgevoerd dat de PRIMARY-status veronderstelt.

Begin elk commandobestand met CLEAR, zodat dit vanuit een "bekende" status start.

Voorbeeld:

```
.CLEAR
```

52 ... CONTINUE

CONTINUE

Dit commando wordt samen met het LOCATE-commando gebruikt. LOCATE en CONTINUE mogen worden gescheiden door andere commando's, waarbij echter bepaalde beperkingen gelden. Zie LOCATE voor meer informatie.

COPY

```
COPY TO <bestand> [ <bereik> ] [ FIELD <veldlijst> ] [ FOR <uitdr> ]
      [ SDF ] [ WHILE <uitdr> ] [ DELIMITED [ WITH <scheidingst> ] ]
```

```
COPY TO <bestand> STRUCTURE [ EXTENDED ]
      [ FIELD <veldlijst> ]
```

Met dit commando wordt de database in USE gekopieerd naar een ander bestand. Het <bestand> kan in dBASE- of System Data Format zijn opgesteld (als de SDF-optie wordt opgegeven).

Als de STRUCTURE-bepaling gespecificeerd wordt, wordt alleen de structuur van een dBASE-bestand in USE naar het "TO"-bestand gekopieerd.

Als na een FIELD-bepaling een lijst met velden gespecificeerd wordt, dan worden naar het "TO"-bestand alleen deze gegevensvelden gekopieerd. Bij COPY TO <bestand> STRUCTURE FIELD <lijst> wordt alleen de structuur van de genoemde velden naar het "TO"-bestand gekopieerd. In beide gevallen bestaat de nieuwe structuur uitsluitend uit de velden die na FIELD opgegeven werden. Zonder FIELD-bepaling worden alle velden gekopieerd.

Als de bepaling SDF aanwezig is, dan wordt het bestand in USE zonder zijn structuur naar een ander bestand gekopieerd. Dit nieuwe bestand is in het ASCII-standaardformat. Zo kan men bestanden creëren voor andere processors dan dBASE. STRUCTURE en SDF sluiten elkaar uit.

Als ook het sleutelwoord DELIMITED in het commando voorkomt, plaatst het systeem alle tekenvelden in het outputbestand tussen aanhalingstekens en worden de velden door komma's gescheiden. Dit is het omgekeerde van een APPEND DELIMITED. Zonder opgave worden bij DELIMITED in het COPY-commando enkele aanhalingstekens als scheidingstekens gebruikt om tekenreeksvelden aan te geven. Bij het WITH-zinsdeel van DELIMITED kan elk teken als scheidingsteken gebruikt worden. Als een "," als scheidingsteken gebruikt wordt, worden de spaties aan het einde van de tekenvelden en aan het begin van numerieke velden verminderd en staan de tekenreeksen niet tussen aanhalingstekens. Het APPEND-commando herkent alleen enkele en dubbele aanhalingstekens.

Als DELIMITED of SDF wordt gebruikt, dan zal de naam van het output-<bestand> als standaard systeemwaarde de uitbreiding .TXT hebben; in alle andere gevallen zijn de uitbreidingsletters .DBF.

Als het "TO"-bestand nog niet bestaat, wordt het alsnog gecreëerd. Als er reeds een bestand met dezelfde naam aanwezig is, wordt dat vernietigd.

54 ... COPY

COPY TO <bestand> STRUCTURE [EXTENDED] [FIELD <veldlijst>]

Met dit commando wordt de structuur van het database-bestand in USE in de vorm van records naar de opgegeven database gekopieerd. De structuur kan daarna tijdens het uitvoeren van het programma onderzocht worden.

Voorbeeld:

```
.USE VTESTDB
.DISP STRU

STRUCTURE FOR FILE:      C:VTESTDB.DBF
NUMBER OF RECORDS:      00008
DATE OF LAST UPDATE:    20/12/83
PRIMARY USE DATABASE

FLD      NAME              TYPE      WIDTH      DEC
001      TEKENS            C          10         0
002      LOG:VELD          L           1         0
003      DEC:VELD          N          10         2
004      GEH:VELD          N          10         0
**TOTAL**
00017

.COPY STRUCTURE EXTENDED TO TEST
00004 RECORDS COPIED

.USE TEST
.DISP STRU

STRUCTURE FOR FILE:      C:TEST.DBF
NUMBER OF RECORDS:      00004
DATE OF LAST UPDATE:    20/12/83
PRIMARY USE DATABASE

FLD      NAME              TYPE      WIDTH      DEC
001      VELD:NAAM         C          10         0
002      VELD:TYPE         C           1         0
003      VELD:LEN          N           3         0
004      VELD:DEC          N           3         0
**TOTAL**
00018

.LIST

00001      TEKENS            C          10         0
00002      LOG:VELD          L           1         0
00003      DEC:VELD          N          10         2
00004      GEH:VELD          N          10         0
```

(Hier kunnen de gegevens [in dit geval een database-structuur] gewijzigd worden).

Voorbeelden:

```
.DISPLAY ALL OFF NAAM,          TST:NR
TIMMERMANS, THEO                1357
KNOOPS, HERMAN                  2468
BERINGS, MARIJKE                3344
SMULDERS, KEES                  6743
PAPAMOROULOS, D                 1011
DE RUITER, JAN-JORIS            6623
DE JONG, ELLEN                  8093
```

```
.DISPLAY STRUCTURE
STRUCTURE FOR FILE:             TEL:BSTND
NUMBER OF RECORDS:              00007
DATE OF LAST UPDATE:            14/12/82
PRIMARY USE DATABASE
```

FLD	NAME	TYPE	WIDTH	DEC
001	NAAM	C	020	
002	TST:NR	C	005	
003	AFDELING	C	010	
TOTAL			00036	

```
.COPY TO DUPE
00007 RECORDS COPIED
```

```
.COPY TO DUPE2 FOR TST:NR < '8000'
00006 RECORDS COPIED
```

```
.USE DUPE2
```

```
.DISPLAY ALL
```

00001	TIMMERMANS, THEO	1357	123/456
00002	KNOOPS, HERMAN	2468	180/103
00003	BERINGS, MARIJKE	3344	264/401
00004	SMULDERS, KEES	6743	190/901
00005	PAPAMOROULOS, D	1011	84/138
00006	DE RUITER, JAN-JORIS	6623	170/430

```
.USE EXAMPLE
```

```
.COPY FIELD NAAM, TST:NR TO DUPE3
00007 RECORDS COPIED
```

.USE DUPE3

STRUCTURE FOR FILE: DUPE3
 NUMBER OF RECORDS: 00007
 DATE OF LAST UPDATE: 20/12/82
 PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
001	NAAM	C	020	
002	TST:NR	C	005	
TOTAL			00036	

.DISPLAY ALL

00001	TIMMERMANS, THEO	1357
00002	KNOOPS, HERMAN	2468
00003	BERINGS, MARIJKE	3344
00004	SMULDERS, KEES	6743
00005	PAPAMOROULOS, D	1011
00006	DE RUITER, JAN-JORIS	6623
00007	DE JONG, ELLEN	8093

.USE EXAMPLE

.COPY NEXT 4 TO DUPE5
 00004 RECORDS COPIED

.USE DUPE5

.DISPLAY ALL

00001	TIMMERMANS, THEO	1357	123/456
00002	KNOOPS, HERMAN	2468	180/103
00003	BERINGS, MARIJKE	3344	264/401
00004	SMULDERS, KEES	6743	190/901

(de beperkte COPY)

.USE BESTEL

.DISP STRUCTURE
 STRUCTURE FOR FILE: BESTEL.DBF
 NUMBER OF RECORDS: 00012
 DATE OF LAST UPDATE: 01/07/82
 PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
001	KLANT	C	020	
002	ART:NR	C	005	
003	AANTAL	N	005	
TOTAL			00031	

.LIST			
00001	TILLAERTS, TINEKE	31415	13
00002	MEYKOORTS, HOKKE	76767	13
00003	CHANG, LEE	11528	44
00004	ADAMS, LOUISE	89793	12
00005	GOOSSENS, GUUS	31415	3
00006	JOOSTEN, DICK	76767	5
00007	SCHILPEROORT, PIET	21828	5
00008	TARAM, CLARA	70296	9
00009	BARNER, THOMAS	31415	6
00010	SPRINGER, NICO	76767	17
00011	LUKE, LUCKY	89793	4
00012	VAN DE WETERING, PAUL	92653	15

.COPY TO DELIM.DAT DELIMITED
00012 RECORDS COPIED

'TILLAERTS, TINEKE	','31415',	13
'MEYKOORTS, HOKKE	','76767',	13
'CHANG, LEE	','11528',	44
'ADAMS, LOUISE	','89793',	12
'GOOSSENS, GUUS	','31415',	3
'JOOSTEN, DICK	','76767',	5
'SCHILPEROORT, PIET	','21828',	5
'TARAM, CLARA	','70296',	9
'BARNER, THOMAS	','31415',	6
'SPRINGER, NICO	','76767',	17
'LUKE, LUCKY	','89793',	4
'KLAASEN, JAN	','92653',	15

58 ... COUNT

COUNT

```
COUNT [<bereik>] [FOR <uitdr>] [TO <geheugenvar>]
      [WHILE <uitdr>]
```

Telt het aantal records in het bestand in USE. Als FOR aanwezig is, worden alleen de records geteld die aan de uitdrukking voldoen. Als TO gegeven is, wordt het getelde aantal in een geheugenvariabele geplaatst. Als de geheugenvariabele nog niet bestaat, wordt deze alsnog gecreëerd. Als DELETED op OFF staat worden gewisse records meegeteld; als DELETED op ON staat worden de gewisse records genegeerd.
dBASE antwoordt met de boodschap:

```
COUNT = xxxxx
```

Voorbeelden:

```
.USE VOORRAAD
```

```
.DISPLAY STRUCTURE
```

```
STRUCTURE FOR FILE:      VOORRAAD
NUMBER OF RECORDS:      00010
DATE OF LAST UPDATE:    23/10/82
PRIMARY USE DATABASE
```

FLD	NAME	TYPE	WIDTH	DEC
001	ART:NR	N	006	
002	CAT:NR	N	003	
003	VKOPER:NR	N	005	
004	BESCHR	C	013	
005	PRIJS:STUK	N	007	002
006	PLAATS	C	005	
007	VOORRADIG	N	004	
008	VERKOCHT	N	004	
009	PRIJS	N	007	002
** TOTAL **			00055	

```
.DISPLAY ALL
```

00001	136928	13	1673	SLEUTEL	7.13	189	9	0	9.98
00002	221679	9	1673	HANDZAAG	5.17	173	4	1	7.98
00003	234561	0	96	PLASTIC BUIS	2.18	27	112	53	4.75
00004	556178	2	873	KATROL	22.19	117	3	0	28.50
00005	723756	73	27	ELEK. KAST	19.56	354	6	1	29.66
00006	745336	13	27	ZEKERINGDOOS	12.65	63	7	2	15.95
00007	812763	2	1673	WERELDBOL	5.88	112	5	2	7.49
00008	876512	2	873	IJZERDRAAD	3.18	45	7	3	4.25
00009	915332	2	1673	CIRKELZAAG	1.32	97	7	3	1.98
00010	973328	0	27	GIETER	0.73	21	17	5	0.99

.COUNT
COUNT = 00010

.COUNT FOR ART:NR > 500000
COUNT = 00007

.COUNT FOR 'ZAAG'\$BESCHR
COUNT = 00002

.GOTO TOP

.COUNT FOR PRIJS:STUK < 10 NEXT 6
COUNT = 00003

.GOTO TOP

.COUNT NEXT 6 FOR PRIJS:STUK < 10
COUNT = 00003

.USE B:INKOPEN

.LIST

00001	BONEN	5	0.75
00002	BROOD	2	0.97
00003	BIEFSTUK	4	3.94
00004	PAPIEREN BORDEN	1	0.86
00005	PLASTIC VORKEN	5	0.42
00006	KROPSLA	2	0.53
00007	KAAS	1	1.96
00008	MELK	2	1.30
00009	HOUTSKOOL	2	0.75

.DISPLAY STRUCTURE

STRUCTURE FOR FILE: B:INKOPEN.DBF
NUMBER OF RECORDS: 00009
DATE OF LAST UPDATE: 10/12/83
PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
001	ITEM	C	020	
002	NR	N	005	
003	PRIJS	N	010	002
TOTAL			00036	

.COUNT TO XX FOR PRIJS > 1
COUNT = 0003

.? XX
3

CREATE

```
.CREATE [ <bestandsnaam> ]  
        [ <bestandsnaam> FROM <BESTANDSNAAM> ]
```

Creëert een nieuw bestand met dBASE-structuur. De gebruiker bepaalt voor dit database-bestand de structuur, de veldnamen en de bestandsnaam.

Als op het commando geen bestandsnaam volgt, krijgt de gebruiker een prompt waarin de <bestandsnaam> gevraagd wordt:

FILENAME:

De gebruiker voert een geldige bestandsnaam in, die geen tekens mag bevatten die onder CP/M een speciale betekenis hebben (zoals B: om schijf eenheid "B" aan te duiden).

dBASE wacht nu op de invoer van de database-structuur. De volgende boodschap verschijnt:

```
ENTER RECORD STRUCTURE AS FOLLOWS:  
FIELD          NAME,TYPE,WIDTH,DECIMAL PLACES  
001
```

De gebruiker voert nu de veldnamen en de bijbehorende structuurgegevens in. Een veldnaam is een tekenreeks van max. 10 tekens, die letters, cijfers en dubbele punten kunnen zijn. Veldnamen moeten met een letterteken beginnen. Er zijn drie veldtypes: tekenreeks, numeriek of logisch. Het veldtype wordt bepaald door één teken:

C – tekenreeks
N – numeriek
L – logisch

"WIDTH" verwijst naar de lengte van het veld. Een tekenreeks kan b.v. 20 tekens lang zijn (m.a.w. de veldlengte is 20). Numerieke gegevens zijn ofwel gehele ofwel decimale getallen. De lengte van gehele getallen is het max. aantal cijfers dat in het veld verwacht wordt. Voor decimale getallen zijn twee lengtes vereist: de eerste is het max. aantal cijfers dat in het decimale getal verwacht wordt (met inbegrip van het decimaalteken); de tweede breedte is het aantal cijfers, dat rechts van het decimaalteken wordt toegestaan. Logische velden zijn altijd slechts 1 teken lang.

Het commando CREATE <bestandsnaam> FROM <BESTANDS-NAAM> creëert een nieuw bestand door de structuur van de records uit het FROM-bestand te lezen. Het FROM-bestand kan geschreven worden door het commando COPY STRUCTURE EXTENDED of door de gebruikelijke gegevens-invoermethodes.

Voorbeeld:

```
.USE TEST
.DISP STRU
```

```
STRUCTURE FOR FILE:      C:TEST.DBF
NUMBER OF RECORDS:      00004
DATE OF LAST UPDATE:    14/12/83
PRIMARY USE DATABASE
```

FLD	NAME	TYPE	WIDTH	DEC
001	VELD:NAAM	C	010	
002	VELD:TYPE	C	001	
003	VELD:LEN	N	003	
004	VELD:DEC	N	003	
TOTAL			00018	

```
.LIST
```

00001	TEKENS	C	10	0
00002	LOG:VELD	L	1	0
00003	DEC:VELD	N	10	2
00004	GEH:VELD	N	10	0

(Hier kunnen de gegevens [die een database-structuur inhouden] gewijzigd worden.)

```
.CREATE TEST2 FROM TEST
```

```
.USE TEST2
.DISP STRU
```

```
STRUCTURE FOR FILE:      C:TEST2.DBF
NUMBER OF RECORDS:      00000
DATE OF LAST UPDATE:    14/12/82
PRIMARY USE DATABASE
```

FLD	NAME	TYPE	WIDTH	DEC
001	TEKENS	C	010	
002	LOG:VELD	L	001	
003	DEC:VELD	N	010	
004	GEH:VELD	N	010	
TOTAL			00032	

```
.CREATE
```

```
FILENAME: VOORBLD
```

```
ENTER RECORD STRUCTURE AS FOLLOWS:
```

```
FIELD NAME,TYPE,WIDTH,DECIMAL PLACES
```

```
001 NAAM,C,20
002 TST:NR,C,5
003 AFDELING,C,10
004 <return >
INPUT DATA NOW? Y
```

62 ... CREATE

RECORD 00001

NAAM: TIMMERMANS, THEO
TST:NR: 1357
AFDELING: 123/456

RECORD 00002

NAAM: KNOOPS, HERMAN
TST:NR: 2468
AFDELING: 180/103

RECORD 00003

NAAM: BERINGS, MARIJKE
TST:NR: 3344
AFDELING: 264/401

RECORD 00004

NAAM: SMULDERS, KEES
TST:NR: 6743
AFDELING: 190/901

RECORD 00005

NAAM: PAPAMOROULOS, D
TST:NR: 1011
AFDELING: 84/138

RECORD 00006

NAAM: < return >

.DISPLAY STRUCTURE

NO DATABASE FILE IN USE, ENTER FILENAME: VOORBLD

STRUCTURE FOR FILE: VOORBLD

NUMBER OF RECORDS: 00005

DATE OF LAST UPDATE: 14/12/83

PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
001	NAAM	C	020	
002	TST:NR	C	005	
003	AFDELING	C	010	
TOTAL			00036	

.DISPLAY ALL

00001	TIMMERMANS, THEO	1357	123/456
00002	KNOOPS, HERMAN	2468	180/103
00003	BERINGS, MARIJKE	3344	264/401
00004	SMULDERS, KEES	6743	190/901
00005	PAPAMOROULOS, D	1011	84/138

DELETE

```
DELETE [<bereik>] [FOR <uitdr>]
          [WHILE <uitdr>]
DELETE FILE <bestandsnaam>
```

Alle records die binnen het <bereik> liggen (en voldoen aan de FOR-uitdrukking indien aanwezig) krijgen een wismarkering. De standaard systeemwaarde voor bereik is alleen het huidige record. Records met een wismarkering worden uit het bestand pas echt verwijderd door een PACK commando. COPY, APPEND en SORT negeren records met een wismarkering. Met RECALL kunt u alle wismarkeringen verwijderen. Records met wismarkering verschijnen nog steeds op het scherm. De wismarkering verschijnt dan als een sterretje tussen het recordnummer en het eerste veld. Zie ook het SET DELETED ON/OFF-commando.

In de tweede vorm wordt het bestand met naam <bestandnaam> van de diskette verwijderd (indien mogelijk) en wordt de ruimte die door dit bestand werd ingenomen vrijgemaakt, zodat het besturingssysteem die opnieuw kan gebruiken. Als <bestandsnaam> echter in USE is kan dat bestand niet gewist worden.

Voorbeelden:

```
.LIST
00001 136928 13 1673 SLEUTEL      7.13 189  9 0  9.98
00002 221679  9 1673 HANDZAAG    5.17 173  4 1  7.98
00003 234561  0  96  PLASTIC BUIS  2.18  27 112 53  4.75
00004 556178  2  873 KATROL      22.19 117  3 0 28.50
00005 723756 73  27  ELEK. KAST  19.56 354  6 1 29.66
00006 745336 13  27  ZEKERINGDOOS 12.65  63  7 2 15.95
00007 812763  2 1673 WERELDBOL   5.88 112  5 2  7.49
00008 876512  2  873 CIRKELZAAG  3.18  45  7 3  4.25
00009 915332  2 1673 VIJL        1.32  97  7 3  1.98
```

```
.DELETE RECORD 2
00001 DELETION(S)
```

```
.5
```

```
.DELETE NEXT 3
00003 DELETION(S)
```

64 ... DELETE

.LIST

00001	136928	13	1673	SLEUTEL	7.13	189	9	0	9.98
00002	*221679	9	1673	HANDZAAG	5.17	173	4	1	7.98
00003	234561	0	96	PLASTIC BUIS	2.18	27	112	53	4.75
00004	556178	2	873	KATROL	22.19	117	3	0	28.50
00005	*723756	73	27	ELEK. KAST	19.56	354	6	1	29.66
00006	*745336	13	27	ZEKERINGDOOS	12.65	63	7	2	15.95
00007	*812763	2	1673	WERELDBOL	5.88	112	5	2	7.49
00008	876512	2	873	CIRKELZAAG	3.18	45	7	3	4.25
00009	915332	2	1673	VIJL	1.32	97	7	3	1.98

.RECALL ALL

00004 RECALL(S)

.LIST

00001	136928	13	1673	SLEUTEL	7.13	189	9	0	9.98
00002	221679	9	1673	HANDZAAG	5.17	173	4	1	7.98
00003	234561	0	96	PLASTIC BUIS	2.18	27	112	53	4.75
00004	556178	2	873	KATROL	22.19	117	3	0	28.50
00005	723756	73	27	ELEK. KAST	19.56	354	6	1	29.66
00006	745336	13	27	ZEKERINGDOOS	12.65	63	7	2	15.95
00007	812763	2	1673	WERELDBOL	5.88	112	5	2	7.49
00008	876512	2	873	CIRKELZAAG	3.18	45	7	3	4.25
00009	915332	2	1673	VIJL	1.32	97	7	3	1.98

.DISP FILES ON B

DATABASE FILES	#	RCDS	LAST UPDATE
INKOPEN .DBF	00007		06/06/82
INKOPEN2 .DBF	00007		05/06/82

.DELETE FILE B:INKOPEN2

FILE HAS BEEN DELETED

.DISPLAY FILES ON B

DATABASE FILES	#	RCDS	LAST UPDATE
INKOPEN .DBF	.00007		06/06/82

DISPLAY

- a. DISPLAY [<bereik >] [FOR <uitdr >] [<lijst uitdr >]
[OFF] [FIELDS <lijst >]
[WHILE <uitdr >]
- b. DISPLAY STRUCTURE
- c. DISPLAY MEMORY
- d. DISPLAY FILES [ON <drive >] [LIKE <masker >]
- e. DISPLAY STATUS

Display (weergave) is de basis van dBASE. Het doel van alle handelingen is per slot van rekening dat de gegevens uit de databases geheel, gedeeltelijk of samengevat op commando op het scherm van de gebruiker verschijnen. Daarvoor dient nu het DISPLAY-commando met zijn vele selectiemogelijkheden.

In vorm a., wordt de database in USE geheel of gedeeltelijk weergegeven. Als <bereik > niet gespecificeerd wordt, en de FOR <uitdr > niet aanwezig is, dan wordt alleen de informatie uit het huidige record getoond. Als <bereik > niet gespecificeerd wordt maar er wel een FOR <uitdr > gegeven is, zullen alle records in de database voor weergave in aanmerking komen. Alle velden worden weergegeven, tenzij <lijst uitdr > aanwezig is. Geldige uitdrukkingen kunnen bestaan uit gegevensvelden, geheugenvariabelen of een willekeurige constante, teken of logische waarde. Het huidige recordnummer wordt bij elke weergegeven regel vermeld, behalve bij keuze van de optie OFF. Als FOR gespecificeerd wordt, zullen alleen de records die aan de voorwaarde achter FOR voldoen worden getoond. Met DISPLAY FIELDS <lijst > kunnen veldnamen worden gebruikt (en weergegeven) die anders dubbelzinnig zouden zijn (zoals bijvoorbeeld STATUS, FILE, STRUCTURE).

Nadat groepen van 15 records weergegeven zijn, gaat DISPLAY pas verder als de gebruiker een toets aanslaat. Zo kan de gebruiker door een lange weergave "bladeren". Het LIST-commando is identiek aan het DISPLAY-commando, maar LIST wacht niet na de weergave van recordgroepen, en de standaard systeemwaarde voor bereik is ALL (alle) records. Met de <escape > -toets worden de commando's LIST en DISPLAY beëindigd.

In vorm b. wordt alleen de structuur van de database in USE weergegeven. In vorm c. worden alle opgegeven geheugenvariabelen weergegeven met de naam van de geheugenvariabele en de daarbij behorende waarde.

In vorm d. kunnen .DBF-bestanden worden weergegeven die op de standaard-drive (of op <drive >) staan, samen met een aantal kenmerken van de database. De LIKE-zin biedt de mogelijkheid ook andere bestandstypes weer te geven. <masker > is van de vorm *.type, waarbij .type kan staan voor .TXT, .FRM, .MEM of een andere tekenreeks van drie letters. Deze bestanden worden op dezelfde manier weergegeven als in het CP/M-commando DIR.

In vorm e. worden alle bestanden weergegeven die in USE zijn, samen met de indexbestanden en alle sleuteluitdrukkingen die gebruikt (USED) of ingesteld (SET) zijn. DISPLAY STATUS toont ook de status van alle SET commando's.

66 ... DISPLAY

Voorbeelden:

```
.USE B:VOORRAAD
```

```
.DISPLAY STRUCTURE
STRUCTURE FOR FILE:      B.VOORRAAD.DBF
NUMBER OF RECORDS:      00008
DATE OF LAST UPDATE:    18/12/82
PRIMARY USE DATABASE
```

FLD	NAME	TYPE	WIDTH	DEC
001	ARTIKEL	C	020	
002	PRIJS	N	010	002
003	ART:NR	C	005	
004	VOORRADIG	N	005	
TOTAL			00041	BYTES

(opmerking: TOTAL bevat altijd 1 extra-byte)

```
.DISPLAY ALL ARTIKEL,ART:NR,PRIJS*VOORRAAD,$(ART:NR,1,2);
FOR PRIJS > 100 .AND. VOORRAAD > 2 OFF
```

BV SCHAAPJESWOL	DI0090	5050,00	DI
BEREVEL & Co	SK0091	220,00	SK
BONTIMPORT	IM0092	2323,00	IM

```
.DISPLAY MEMORY
```

```
KLANT:NAM (C)  DANGELMEYER, PATRICIA
BUDGET (N)    123456.70
EF:STATUS (L)  .T.
**TOTAL**    03 VARIABLES USED      00027 BYTES USED
```

```
.DISPLAY FILES ON B: LIKE *.FRM
TEST .FRM ADMIN .FRM BESTEL.FRM
```

```
.DISPLAY FILES
```

DATABASE FILES	# RCDS	LAST UPDATE
TEST .DBF	00077	00/00/00
ADRECS .DBF	00073	09/12/83
HISTSTR .DBF	00000	26/03/84
TMPADMIN .DBF		

NOT A dBASE II DATABASE

Het laatste .DBF-bestand uit bovenstaande lijst is het bestand dat geen dBASE database is.

Hier zijn alleen enkele typische voorbeelden van DISPLAY gegeven; bij de andere commando's vindt u nog meer voorbeelden.

DO

- a. DO <bestand>
- b. DO WHILE <uitdr>
<opdrachten>
ENDDO
- c. DO CASE

In vorm a. wordt het bestand <file> geopend en gelezen. Het bestand is in dit geval een COMMANDOBESTAND. Zo'n bestand bestaat volledig uit dBASE-commando's. De input wordt net zo geïnterpreteerd en uitgevoerd als commando's die vanaf het toetsenbord ingevoerd zouden worden. DO's kunnen tot 16 niveaus genest worden (commando-bestanden kunnen nl. DO-commando's bevatten, waarmee andere commandobestanden worden opgeroepen). Een commandobestand houdt op bij zijn end-of-file, of bij het commando RETURN. Als het huidige commandobestand is opgeroepen door een ander commandobestand, neemt het commandobestand van het hogere niveau de besturing weer over. Als het systeem tijdens het uitvoeren van een commandobestand een CANCEL-commando aantreft, worden alle commandobestanden gesloten en verloopt de besturing verder via het toetsenbord.

In vorm b. worden de opdrachten (statements) na DO uitgevoerd als de uitdrukking <uitdr> logisch waar is. Na een ENDDO-instructie gaat dBASE over naar de instructie die op de ENDDO-instructie volgt, in het andere geval worden alle opdrachten opnieuw uitgevoerd.

Opmerking: <opdrachten> verwijst naar volledige opdrachten. De DO WHILE-opdracht wordt beëindigd door een ENDDO. De opdrachten moeten zorgvuldig genest worden: als er een IF "binnenin" een DO WHILE staan, mag de ENDDO niet voor de ENDIF komen. Zie hoofdstuk 9.2. regel 8 voor meer informatie.

Voorbeelden:

```
DO CREDBET

DO WHILE .NOT. EOF
  DISPLAY NAME
  .
  .
  .
SKIP
ENDDO
```

CASE is een uitbreiding van het DO-commando en heeft dezelfde vorm als het bovenstaande commando. Het aantal CASE-termen in een DO CASE is onbeperkt. De zinsnede OTHERWISE is facultatief.

DO CASE is een gestructureerde procedure. De afzonderlijke CASE's in de constructie kunnen worden beschouwd als de uitzonderingen op de regel die OTHERWISE definieert. Als één bepaalde voorwaarde op een speciale manier verwerkt moet worden, dan zou die voorwaarde een CASE zijn en alle andere OTHERWISE-voorwaarden. OTHERWISE kan dan ook worden beschouwd als de standaardwaarde. Zie het eerste voorbeeld op de volgende bladzijde.

De manier waarop dBASE de DO CASE-constructie behandelt kan het best vergeleken worden met een reeks IF's. M.a.w. dBASE zal de DO CASE als een reeks IF-ENDIF's uitvoeren.

DO CASE	IF ITEM = 'SINAASAPPELEN'
CASE ITEM = 'SINAASAPPELEN'	opdrachten
opdrachten	ELSE
CASE ITEM = 'APPELEN'	IF ITEM = 'APPELEN'
opdrachten	opdrachten
OTHERWISE	ELSE
opdrachten	opdrachten
ENDCASE	ENDIF
	ENDIF

dBASE zal dus de <uitdr> in de afzonderlijke CASE's onderzoeken en zo gauw de uitdrukking waar is, wordt de opdracht die erop volgt uitgevoerd. Als dBASE daarna een volgende zin tegenkomt die met CASE begint, dan wordt doorgeslagen naar ENDCASE. Dit betekent dat als er meer dan één uitdrukking waar is, alleen de eerste opdracht uitgevoerd wordt.

Als de OTHERWISE-bepaling opgenomen is en geen enkele CASE waar is, dan worden de <opdrachten> in de OTHERWISE-bepaling uitgevoerd. Als er geen OTHERWISE-bepaling is en geen enkele CASE waar is, dan verlaat dBASE de DO CASE zonder ook maar één <opdracht> uit te voeren.

Opdrachten die tussen de "DO CASE" en de eerste "CASE" in staan, worden niet uitgevoerd.

Voorbeelden:

```

DO CASE
  CASE MERK = "VOLVO"
    < opdrachten > die VOLVO verwerken
  CASE MERK = "OPEL"
    < opdrachten > die OPEL verwerken
  CASE MERK = "AUDI"
    < opdrachten > die AUDI verwerken
  OTHERWISE
    < opdrachten > die alle andere automerken verwerken
ENDCASE

```

In bovenstaand voorbeeld hebben alle uitdrukkingen betrekking op dezelfde veldnaam. Dit hoeft echter niet zo te zijn. In een < uitdr > kan van alles staan en de verschillende CASE's hoeven niet per se met elkaar in verband te staan.

```

DO CASE
  CASE VANDAAG = "MAANDAG"
    < opdrachten > voor MAANDAG
  CASE WEER = "REGEN"
    < opdrachten > voor REGEN
  CASE STAD = "AMSTERDAM"
    < opdrachten > voor AMSTERDAM
ENDCASE

```

Als het nu een regenachtige maandag in Amsterdam is, dan wordt toch alleen de CASE voor MAANDAG uitgevoerd.

CASE's hoeven niet altijd tekenreeksen te zijn, zoals in de twee voorgaande voorbeelden. Elke uitdrukking is mogelijk.

```

DO CASE
  CASE X = Y + Z
    < opdrachten > voor optelling
  CASE .NOT.A
    < opdrachten > voor Booleaanse logica
  CASE "A"$"ABCDEF"
    < opdrachten > voor reekslogica
  OTHERWISE
    < opdrachten >
ENDCASE

```

EDIT

EDIT [n]

Met het EDIT-commando kan de gebruiker de inhoud van de database selectief veranderen. EDIT kan op verschillende manieren gebruikt worden en werken, afhankelijk van het feit of dBASE in full-screen-modus staat of niet (zie SET SCREEN-commando).

Als dBASE in de full-screen-modus staat, kan een record met "EDIT" of "EDIT n" gewijzigd worden (n verwijst naar het record dat gewijzigd moet worden). Als n niet gegeven is, vraagt dBASE naar de coördinaten van het te wijzigen record. In een andere modus gebeurt dit op dezelfde manier, zodra het recordnummer echter gegeven wordt is de full-screen werking weer beschikbaar. Zie hoofdstuk 8, full-screen werking, voor een beschrijving van controltoetsen en cursorbeweging.

Als een SET FORMAT TO <bestand> in werking is, dan gebruikt EDIT de @-commando's uit het formatbestand om het scherm in te delen en het toevoegen van gegevens te besturen. Anders geeft EDIT alle velden in tabelvorm weer.

Als het EDIT-commando niet in de full-screen-modus gebruikt wordt, antwoordt dBASE met:

COORD:

De gebruiker voert dan de coördinaten in van het te wijzigen gegevensveld en (eventueel) de nieuwe waarde. De coördinaten van het gegevensveld zijn: het recordnummer en het veldnummer (of de veldnaam). Als een nieuwe waarde gegeven wordt, zal dBASE de inhoud van het bepaalde veld door de nieuwe waarde vervangen. Als er geen nieuwe waarde gegeven wordt geeft dBASE de inhoud van het veld weer en vraagt aan de gebruiker of er veranderingen aangebracht moeten worden. Als dat niet zo is drukt de gebruiker op <return>, zodat er aan de inhoud van het veld niets veranderd wordt. Of er nu wijzigingen worden aangebracht of niet, dBASE vraagt de gebruiker naar de coördinaten van het volgende record met de prompt "COORD:"

Na invoering van het eerste coördinatenpaar, kan de gebruiker één van beide coördinaat-waarden weglaten en gebruikt dBASE voor die coördinaat de vorige waarde. De gebruiker verlaat de EDIT-modus door als antwoord op de COORD-vraag, <return> in te toetsen.

Het gehele gegevensveld kan worden gewist door bij verschijnen van de boodschap CHANGE?, <control>-Y en <return> in te toetsen. Zo kan een veld helemaal opnieuw ingevoerd worden. Het wijzigen van een gegevensveld kan worden afgebroken door <control>-Q in te toetsen, waardoor alle aangebrachte wijzigingen genegeerd worden en het gegevensveld zijn oorspronkelijke inhoud terugkrijgt.

Als een geïndexeerd bestand met EDIT gewijzigd wordt en de indexbepaling gebruikt is, zal dBASE de index aanpassen bij verandering van de sleutel. Als er meerdere geïndexeerde bestanden met de database in verband staan, worden de niet bij USE genoemde bestanden door EDIT niet aangepast.

Voorbeelden:

```
.USE INKOPEN
```

```
.DISPLAY STRUCTURE
STRUCTURE FOR FILE:   INKOPEN
NUMBER OF RECORDS:   00006
DATE OF LAST UPDATE: 03/07/83
PRIMARY USE DATABASE
```

FLD	NAME	TYPE	WIDTH	DEC
001	ITEM	C	020	
002	AANTAL	N	005	
003	PRIJS	N	010	002
TOTAL			00036	

```
.LIST
00001 BONEN (BLIK)           5      0.69
00002 BROOD                 2      0.89
00003 BIEFSTUK             4      3.59
00004 KROPSLA              1      0.49
00005 MELK (1 LITER)       2      1.19
00006 HOUTSKOOL            1      0.69
```

```
.EDIT
COORD:      5,ITEM, MELK (1/2 LITER)
```

```
COORD:      2,1
```

```
ITEM: BROOD
```

```
CHANGE?     OD
TO          DEN
```

```
ITEM: BRODEN
CHANGE?     <return>
COORD:      6,1
```

```
ITEM: HOUTSKOOL
CHANGE?     OL
TO          OL, 5 ZAKKEN
```

72 ... EDIT

ITEM: HOUTSKOOL, 5 ZAKKEN

CHANGE? <return>

COORD: ,2

NO: 1

TO: 2

COORD: 4

NO: 1

TO: 2

COORD: <return>

.LIST

00001	BONEN (BLIK)	5	0.69
00002	BRODEN	2	0.89
00003	BIEFSTUK	4	3.59
00004	GROENTE	2	0.49
00005	MELK (1/2 LITER)	2	1.19
00006	HOUTSKOOL, #5 ZAKKEN	2	0.69

(Ook met het volgende gedeelte van een commandobestand zou men op selectieve wijze een database kunnen veranderen. De "&" is essentieel om dit commando te laten werken; hiermee wordt de reeks die met ACCEPT werd aangenomen, omgezet in cijfers die EDIT kan herkennen.)

STORE '1' TO X

DO WHILE X <> '0'

ACCEPT "Voer recordnummer in" TO X

EDIT &X

ENDDO

EJECT

EJECT

Met dit commando geeft de printer een paginasprong bij SET PRINT ON of bij SET FORMAT TO PRINT. Als u het commando gebruikt om een pagina rechtstreeks te formatteren, dan stelt het EJECT-commando de regel- en kolomregisters op nul. Zie ook het commando SET EJECT ON/OFF.

Voorbeeld:

.EJECT

74 ... ERASE

ERASE

ERASE

Dit commando wist het scherm en plaatst de cursor in de linkerbovenhoek van het scherm. Als het @-commando gebruikt wordt en SET SCREEN ON in werking is, dan wist ERASE alle GET's en PICTURE's uit het geheugen van voorgaande @-commando's.

Voorbeeld:

.ERASE

FIND

FIND <tekenreeks> of FIND ' <tekenreeks> '

Met dit commando zoekt dBASE het eerste record in een geïndexeerde database (in USE) waarvan de sleutel gelijk is aan <tekenreeks>. FIND kan records in een geïndexeerde database snel lokaliseren. De FIND-tijd op een diskette-systeem bedraagt gemiddeld 2 seconden.

FIND werkt alleen met databases die van tevoren geïndexeerd zijn (zie beschrijving van het INDEX-commando). Als het INDEX-commando als sleutel een tekenreeks-uitdrukking gebruikt heeft, dan werkt FIND ook als u alleen de eerste paar tekens van de sleutel opgeeft. FIND levert dan het eerste record waarvan de sleutel identiek is aan de <tekenreeks> voor het aantal opgegeven tekens. Voorbeeld: de opdracht 'FIND SMI' zal een record met als sleutel 'SMITS, JAN' vinden, als er tenminste geen andere sleutels zijn die met 'SMI' beginnen en in de index voor 'SMITS, JAN' komen. FIND vindt altijd alleen het eerste record waarvan de sleutel dezelfde is als de <tekenreeks>. Zelfs wanneer de recordpointer verderop in het bestand geplaatst wordt, zal een FIND-commando met dezelfde sleutel opnieuw het EERSTE record opzoeken.

Als de index op een numerieke sleutel gebaseerd is, dan is het opgezochte record het eerste record waarvan de sleutel rekenkundig gelijk is aan het FIND-argument.

Opmerking: Voor indexen met sleutels die zowel tekens als cijfers bevatten, mag het FIND-item als een tekenreeks met of zonder aanhalingstekens als scheidingstekens geschreven worden. Alleen als de sleutel vooraan spaties bevat moet de tekenreeks tussen aanhalingstekens staan. In dat geval moet tussen de aanhalingstekens het juiste aantal spaties worden gegeven die vooraan stonden.

Als u een geheugenvariabele in het FIND-argument wenst, dan geeft u die op na het FIND-commando door middel van een &-macro-substitutie. Bijvoorbeeld: FIND &NAME waarbij NAME een tekenreeks-geheugenvariabele is. Numerieke geheugenvariabelen moeten eerst door middel van een STR-functie in reeksen worden omgezet, anders kunnen ze niet door een "macro" worden vervangen. Zie hoofdstuk 5 voor meer informatie over macro's.

Zodra een record in een database door middel van het FIND-commando gelokaliseerd is, kan het net als elk ander database-record verwerkt worden. Het kan dus worden bekeken, gewijzigd, of gebruikt in berekeningen enz. dBASE-commando's die een database doorlopen (b.v. LIST, REPORT, COPY, enz.) verwerken eerst het gevonden record en gaan dan naar het volgende record uit de reeks, naar volgorde van de sleutel.

Als er geen record bestaat waarvan de sleutel gelijk is aan de tekenreeks, verschijnt het bericht "NO FIND" op het scherm en levert de recordnummer-functie "#" de waarde 0.

76 ... FIND

Als u een tweede record met dezelfde sleutel wenst, dan zult u een SKIP of een LOCATE FOR < uitdr > moeten gebruiken. SKIP weet niet wanneer er geen overeenkomende records meer zijn; met LOCATE kunnen nog andere overeenkomende records gevonden worden (voor zover de sleutel in de uitdrukking werd gebruikt).

Bij SET EXACT ON zal FIND alleen iets vinden als de GEHELE sleutel teken voor teken identiek is (alleen het aantal spaties achteraan mag verschillen).

Bij SET DELETE ON zal FIND geen records vinden die met het DELETE-commando een wismarkering gekregen hebben.

Voorbeelden:

```
.USE INKOPEN INDEX INKINDEX
```

```
.LIST
```

00003	Biefstuk	4	4.33
00001	Bonen	5	0.75
00002	Brood	2	1.06
00006	Groente	2	0.53
00009	Houtskool	2	0.75
00007	Kaas	1	1.96
00008	Melk	2	1.30
00004	Papieren borden	1	0.94
00005	Plastic vorken	5	0.42

```
.FIND Brood
```

```
.DISPLAY
```

00002	Brood	2	1.06
-------	-------	---	------

```
.DISPLAY NEXT 3
```

00002	Brood	2	1.06
00006	Groente	2	0.53
00009	Houtskool	2	0.75

```
.FIND P
```

```
.DISPLAY
```

00004	Papieren borden	1	0.94
-------	-----------------	---	------

```
.FIND Plas
```

```
.DISPLAY
```

00005	Plastic vorken	5	0.42
-------	----------------	---	------

```
.FIND P
```

```
.DISPLAY
```

00004	Papieren borden	1	0.94
-------	-----------------	---	------

FIND werkt in een bestand dat met een meervoudige sleutel geïndexeerd is als deze sleutel alle spaties bevat die tussen de verschillende gedeelten van de sleutel staan.

```
.LIST
00001 De Prinses op de erwt Andersen, H.C. IMB001 29/02/04
00005 Het Lelijke Eendje Andersen, H.C. IBM002 25/03/06
00003 Hoe leer ik ermee leven Dalen, D. van ML001 01/01/30
00007 Fietsen door Nederland Gazelle, P. LS001 12/01/73
00008 Italiaans op reis Giovani, J. KG001 08/04/44
00002 Tuinieren Groen, B. URF001 12/11/23
00006 Hyperventilatie Hart, H. TT001 05/07/42
00009 Sauzen Kok, P. LS002 04/10/73
00004 Jip en Janneke Schmidt, A.M.G. IL001 04/01/00
00010 De Waddeneilanden Veen, A. van het KG002 06/09/42
```

```
.FIND "Andersen, H.C. IMB002"
```

```
.DISP
00005 Het Lelijke Eendje Andersen, H.C. IBM002 25/03/06
```

```
.FIND "Dalen, D. van"
```

```
.DISP
00003 Hoe leer ik ermee leven Dalen, D. van ML001 01/01/30
```

```
.FIND "Kok, P. LS002"
```

```
.DISP
00009 Sauzen Kok, P. LS002 04/10/73
```

78 ... GO of GOTO

GO of GOTO

- a. GOTO RECORD <n>
- b. GOTO TOP
- c. GOTO BOTTOM
- d. <n>
- e. GOTO <geheugenvar>

Met dit commando wordt de recordpointer van een database op een andere plaats gezet.

Bij a. en d. wordt de recordpointer op recordnummer <n> gezet. Geval d. is een verkorte schrijfwijze van a.

Bij b. en c. wordt het bestand in USE voor- of achteruit "gespoeld" (TOP/BOTTOM) en wordt de recordpointer op het eerste/laatste record in het bestand gezet. Als het bestand in USE geïndexeerd is, dan hoeft het eerste/laatste record niet altijd het eerste/laatste record in de database te zijn; het is wel het eerste/laatste record volgens de sleutel die gebruikt werd om de database te indexeren.

Met vorm e. kan de pointer op een recordnummer worden geplaatst dat in een geheugenvariabele is opgeslagen.

Voorbeelden:

```

.USE INKOPEN

.GOTO RECORD 6
6

.DISPLAY
00006 GROENTE          2          0.53

.GOTO TOP

.DISPLAY
00001 BONEN           5          0.75

.GOTO BOTTOM

.DISPLAY
00009 HOUTSKOOL       2          0.75

.LIST
00001 BONEN           5          0.75
00002 BROOD           2          0.97
00003 BIEFSTUK        4          3.94
00004 PAPIEREN BORDEN 1          0.86
00005 PLASTIC VORKEN  5          0.42
00006 GROENTE         2          0.53
00007 KAAS            1          1.96
00008 MELK            2          1.30
00009 HOUTSKOOL       2          0.75

.STORE 4 TO RECORDNR
4

.GOTO RECORDNR

.DISP
00004 PAPIEREN BORDEN 1          0.86

```

80 ... HELP

HELP

HELP <commando>

Het HELP-commando geeft een kort overzicht van het opgegeven dBASE-commando en de syntaxis ervan.

Dit commando zoekt op de schijf het bestand DBASEMSG.TXT. Dit is een bestand van het CP/M-type dat met een tekstverwerkingsprogramma kan worden gewijzigd. Het dient om de gebruiker informatie te geven.

Voorbeeld:

.HELP CREATE

IF

```

IF < uitdr >
  < opdrachten >
[ELSE
  < opdrachten > ]
ENDIF

```

Met het IF-commando kunnen andere opdrachten, die meestal in een commandobestand staan, voorwaardelijk worden uitgevoerd. Als de < uitdr > waar is, dan worden de opdrachten na IF uitgevoerd. Als de uitdrukking niet-waar is, dan worden de opdrachten na ELSE uitgevoerd. Als er geen ELSE wordt opgegeven, dan slaat het systeem alle opdrachten over tot het een ENDIF aantreft. IF-commando's kunnen tot op elk niveau genest worden.

Opmerking: < opdrachten > verwijst hier naar volledige commando's. Het IF-commando begint met een IF en eindigt met een ENDIF. De opdrachten moeten goed genest zijn; een IF met een DO WHILE in het pad voor waar (of niet-waar) mag niet voor de DO WHILE eindigen. Zie hoofdstuk 9.8 Regel 8 voor meer informatie.

Voorbeelden:

```

IF STATUS = 'GETROUWD'
  DO GKOSTEN
ELSE
  DO VKOSTEN
ENDIF

IF X = 1
STORE POSTNR + WOONPLAATS TO ADRESLL
ENDIF

```

INDEX

INDEX ON <uitdrukking> TO <indexbestandsnaam>
INDEX

Met het INDEX-commando wordt het bestand in USE geïndexeerd op de <uitdrukking>. De <uitdrukking> noemt men de "sleutel". Dit betekent dat dBASE een bestand opbouwt (het <indexbestand>) met pointers die verwijzen naar de records in het USE-bestand. Het indexbestand is zo opgebouwd dat de USE-database voor de verdere bewerkingen op de sleutel gesorteerd lijkt te zijn. Het bestand in USE blijft echter structureel onveranderd. Het sorteren gebeurt in stijgende volgorde. Een dalende sortering kan gebeuren op een numerieke uitdrukking. Verderop volgt een voorbeeld.

Indexering maakt dat de records uit een database snel kunnen worden teruggevonden, door de sleutel geheel of gedeeltelijk in het FIND-commando (zie FIND) te specificeren. Een database hoeft alleen maar geïndexeerd te worden als dat de verdere bewerkingen kan versnellen of vereenvoudigen. Een geïndexeerde database kan later met of zonder de indexering gebruikt worden.

Vaak hoeft U een INDEX-commando voor een bepaald bestand slechts één keer te geven. Met een APPEND-commando bijvoorbeeld wordt het indexbestand telkens als er nieuwe records worden toegevoegd automatisch bijgewerkt.

Als een geïndexeerde database opnieuw gebruikt wordt (tijdens een latere sessie of later in dezelfde sessie waarin de oorspronkelijke INDEXering werd uitgevoerd), dan moet een speciale vorm van het USE-commando worden gebruikt (nl. USE <database-bestandsnaam> INDEX <indexbestandsnaam>).

Voor elke database kan een onbeperkt aantal indexbestanden worden opgebouwd, maar met de APPEND, EDIT, REPLACE, READ of BROWSE-commando's worden alleen de indexbestanden in USE automatisch bijgewerkt.

Een geïndexeerd bestand kan geschoond worden met het PACK-commando waarbij de database en het indexbestand op de juiste manier worden bijgewerkt. Als er echter meer dan één indexbestand bij de gePACKte database hoort, dan moet de database opnieuw op die sleutels geïndexeerd worden.

Let op: De TRIM-functie mag niet als een deel van een indexsleutel gebruikt worden. Als de functies \$ of STR in een sleutel (of als de sleutel zelf) worden gebruikt, dan moeten hun lengte-parameters constanten zijn (géén variabelen of uitdrukkingen). Voorbeeld: INDEX ON \$(NAAM,N,5)+STR(BEDRAG,5) TO IDXBEST is geldig, maar INDEX ON \$(NAAM,N,N+5)+STR(BEDRAG,LENGTEVAR) TO IDXBEST is ongeldig.

Het commando INDEX zónder parameters indexeert het lopend record naar het indexbestand in USE. Hiermee kan een index worden opgebouwd, waarin niet alle records uit een database zijn opgenomen. Zie ook het commando REINDEX.

VOORBEELDEN:

.USE INKOPEN

.LIST

00001	Bonen	5	0.75
00002	Brood	2	1.06
00003	Biefstuk	4	4.33
00004	Papieren borden	1	0.94
00005	Plastic vorken	5	0.42
00006	Groente	2	0.53
00007	Kaas	1	1.96
00008	Melk	2	1.30
00009	Houtskool	2	0.75

.DISPLAY STRUCTURE

STRUCTURE FOR FILE: INKOPEN.DBF
 NUMBER OF RECORDS: 00009
 DATE OF LAST UPDATE: 03/07/83
 PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
001	ITEM	C	020	
002	NR	N	005	
003	PRIJS	N	010	002
TOTAL			00036	

Aanmaken van indexbestand INKINDEX:

.INDEX ON ITEM TO INKINDEX

Nu lijsten in indexvolgorde:

.LIST

00003	Biefstuk	4	4.33
00001	Bonen	5	0.75
00002	Brood	2	1.06
00006	Groente	2	0.53
00009	Houtskool	2	0.75
00007	Kaas	1	1.96
00008	Melk	2	1.30
00004	Papieren borden	1	0.94
00005	Plastic vorken	5	0.42

Na indexeren werkt het commando FIND:

.FIND Melk

.DISPLAY

00008	Melk	2	1.30
-------	------	---	------

84 ... INDEX

.FIND Bo

.DISPLAY

00001 Bonen	5	0.75
-------------	---	------

.SKIP

RECORD: 00002

.DISPLAY

00002 Brood	2	1.06
-------------	---	------

.SKIP - 1

RECORD: 00001

.DISPLAY

00001 Bonen	5	0.75
-------------	---	------

Het commando USE zonder meer gebruikt het indexbestand niet

.USE INKOPEN

.LIST

00001 Bonen	5	0.75
00002 Brood	2	1.06
00003 Biefstuk	4	4.33
00004 Papieren borden	1	0.94
00005 Plastic vorken	5	0.42
00006 Groente	2	0.53
00007 Kaas	1	1.96
00008 Melk	2	1.30
00009 Houtskool	2	0.75

Andere vorm van commando USE gebruikt het indexbestand wel

.USE INKOPEN INDEX INKINDEX

.LIST

00003 Biefstuk	4	4.33
00001 Bonen	5	0.75
00002 Brood	2	1.06
00006 Groente	2	0.53
00009 Houtskool	2	0.75
00007 Kaas	1	1.96
00008 Melk	2	1.30
00004 Papieren borden	1	0.94
00005 Plastic vorken	5	0.42

.USE BOEKEN
.DISP STRU

STRUCTURE FOR FILE: BOEKEN.DBF
NUMBER OF RECORDS: 00010
DATE OF LAST UPDATE: 18/10/82
PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
001	TITEL	C	025	
002	AUTEUR	C	015	
003	CAT:NUM	C	006	
004	PUBL:DAT	C	008	
TOTAL			00055	

.INDEX ON AUTEUR + CAT:NUM TO BOEKEN
00010 RECORDS INDEXED

.LIST

00001	De Prinses op de erwt	Andersen, H.C.	IMB001	02/09/04
00005	Het Lelijke Eendje	Andersen, H.C.	IBM002	09/12/06
00003	Hoe leer ik ermee leven	Dalen, D. van	ML001	01/01/30
00007	Fietsen door Nederland	Gazelle, P.	LS001	12/01/73
00008	Italiaans op reis	Giovanni, J.	KG001	08/04/44
00002	Tuinieren	Groen, B.	URF001	12/11/23
00006	Hyperventilatie	Hart, H.	TT001	05/07/42
00009	Sauzen	Kok, P.	LS002	04/04/73
00004	Jip en Janneke	Schmidt, A.M.G.	IL001	04/01/00
00010	De Waddeneilanden	Veen, A. van het	KG002	06/12/42

Voorbeeld van sortering in dalende volgorde:

.LIST

00001	GRAS MAAIEN	5
00002	HOND UITLATEN	9
00003	KAT UITLATEN	1
00004	GRAS SPROEIEN	7
00005	SCHILDEREN	4
00006	FIETS REPAREREN	6

.INDEX ON - VOORRANG TO WEEKPLAN
00006 RECORDS INDEXED

.LIST

00002	HOND UITLATEN	9
00004	GRAS SPROEIEN	7
00006	FIETS REPAREREN	6
00001	GRAS MAAIEN	5
00005	SCHILDEREN	4
00003	KAT UITLATEN	1

INPUT

INPUT ["<tekenreeks>"] TO <geheugenvar>

Met deze constructie kunnen waarden van uitdrukkingen in geheugenvariabelen worden opgeslagen en kan in commandobestanden de gebruiker om gegevensinvoer gevraagd worden. Indien nodig wordt de <geheugenvar> gecreëerd en wordt de uitdrukking hierin opgeslagen. Als een <tekenreeks> aanwezig is, wordt deze op het scherm weergegeven als een prompt voordat de invoer aanvaard wordt.

Het type <geheugenvar> wordt gekozen in overeenstemming met de ingevoerde gegevens. Als er een tekenreeks met scheidingstekens wordt ingevoerd, dan is de <geheugenvar> van het type teken. Als er een numerieke uitdrukking wordt ingevoerd, dan is de <geheugenvar> van het numerieke type. Als een T of Y (voor True of Yes) wordt ingevoerd, dan is de <geheugenvar> een logische variabele met de waarde TRUE (waar); als een F of N (voor False of No) wordt ingevoerd, dan is de <geheugenvar> een logische variabele met de waarde FALSE (niet-waar). Met de functie TYPE kan het type van de ingevoerde gegevens expliciet bepaald worden. Om de promptreeks te begrenzen kunnen enkele of dubbele aanhalingstekens gebruikt worden, de tekens aan het begin en aan het einde moeten echter wel dezelfde zijn.

INPUT zou alleen gebruikt moeten worden om numerieke en logische gegevens in te voeren. Om karakterreeksen in te voeren kunt u beter het ACCEPT-commando gebruiken.

Voorbeelden:

```
.INPUT TO X
:3
3
```

```
.INPUT TO Z
:23/17.000 + X
4.352
```

```
.INPUT 'VRAAG GEBRUIKER NAAR INVOER' TO Q
VRAAG GEBRUIKER NAAR INVOER: 12345
12345
```

```
.INPUT 'VOER T IN VOOR O.K.' TO LOG
VOER T IN VOOR O.K.: T
.T.
```

.INPUT "VOER EEN REEKS IN" TO TEKENS
VOER EEN REEKS IN: 'EEN TEKENREEKS MOET TUSSEN
AANHALINGSTEKENS STAAN'
EEN TEKENREEKS MOET TUSSEN AANHALINGSTEKENS STAAN

.DISP MEMO
X (N) 3
Z (N) 4.352
Q (N) 12345
LOG (L) .T.
TEKENS (C) EEN TEKENREEKS MOET TUSSEN
AANHALINGSTEKENS STAAN
TOTAL 05 VARIABLES USED 00069 BYTES USED

.INPUT 'VOER EEN LOGISCHE WAARDE IN' TO LOG2
VOER EEN LOGISCHE WAARDE IN :Y
.T.

INSERT**INSERT [BEFORE] [BLANK]**

Met dit commando kunnen records in het midden van een database worden ingelast. De records kunnen alleen één voor één worden ingelast.

De term BEFORE wordt gebruikt om iets in te lassen vóór het record waarnaar de pointer nu verwijst. Zonder BEFORE wordt het nieuwe record direct achter het huidige ingelast. Net als bij de commando's APPEND en CREATE zal dBase om de invoer van waarden verzoeken, behalve als het woord BLANK aanwezig; in dat geval wordt een leeg record ingelast.

Bij SET CARRY ON wordt de informatie uit het vorige record overgenomen in het nieuwe record.

Als een SET FORMAT TO <bestand> in werking is, dan gebruikt INSERT de @-commando's uit het formatbestand om het scherm te formatteren, waarbij full-screen besturing en validering van de gegevens mogelijk is. Anders brengt INSERT de velden in tabelvorm op het scherm.

INSERTen in grote niet-geïndexeerde databases kan heel lang duren: vermijd dit tenzij het echt nodig is. In een geïndexerd bestand, van welke omvang dan ook, heeft INSERT precies dezelfde werking als APPEND.

Voorbeelden:

```
.USE INKOPEN
```

```
.LIST
```

00001	BONEN # 303 BLIK	5	0.69
00002	BRODEN	2	0.89
00003	BIEFSTUK	4	3.59
00004	GROENTE	2	0.49
00005	MELK (1/2 LIT.)	2	1.19
00006	HOUTSKOOL, 5# ZAKKEN	2	0.69

```
.GOTO RECORD 4
```

```
.INSERT
```

```
RECORD 00005
```

```
ITEM:      KAAS
```

```
AANTAL:    1
```

```
PRIJS:     1.79
```

```
.LIST
```

00001	BONEN # 303 BLIK	5	0.69
00002	BRODEN	2	0.89
00003	BIEFSTUK	4	3.59
00004	GROENTE	2	0.49
00005	KAAS	1	1.79
00006	MELK (1/2 LIT.)	2	1.19
00007	HOUTSKOOL, 5# ZAKKEN	2	0.69

.GOTO RECORD 4

.INSERT BEFORE

RECORD 00004

ITEM: PAPIEREN BORDEN
 AANTAL: 1
 PRIJS: .79

.LIST

00001	BONEN # 303 BLIK	5	0.69
00002	BRODEN	2	0.89
00003	BIEFSTUK	4	3.59
00004	PAPIEREN BORDEN	1	0.79
00005	GROENTE	2	0.49
00006	KAAS	1	1.79
00007	MELK (1/2 LIT.)	2	1.19
00008	HOUTSKOOL, 5# ZAKKEN	2	0.69

.4

.DISPLAY

00004	PAPIEREN BORDEN	1	0.79
-------	-----------------	---	------

.INSERT BLANK

.LIST

00001	BONEN # 303 BLIK	5	0.69
00002	BRODEN	2	0.89
00003	BIEFSTUK	4	3.59
00004	PAPIEREN BORDEN	1	0.79
00005			
00006	GROENTE	2	0.49
00007	KAAS	1	1.79
00008	MELK (1/2 LIT.)	2	1.19
00009	HOUTSKOOL, 5# ZAKKEN	2	0.69

.5

.REPLACE ITEM WITH 'PLASTIC VORKEN', AANTAL WITH 5,
 PRIJS WITH .39

00001 REPLACEMENT(S)

.LIST

00001	BONEN # 303 BLIK	5	0.69
00002	BRODEN	2	0.89
00003	BIEFSTUK	4	3.59
00004	PAPIEREN BORDEN	1	0.79
00005	PLASTIC VORKEN	5	0.39
00006	GROENTE	2	0.49
00007	KAAS	1	1.79
00008	MELK (1/2 LIT.)	2	1.19
00009	HOUTSKOOL, 5# ZAKKEN	2	0.69

JOIN

JOIN TO <bestand> FOR <uitdrukking> [FIELDS <veldlijst>]

Dit is een van de krachtigste commando's van dBASE. Met dit commando kunnen wanneer aan een bepaalde voorwaarde voldaan wordt, twee databases worden samengevoegd tot een derde database.

De twee databases die hiervoor gebruikt worden zijn de primaire en secundaire USE-bestanden. Eerst wordt het commando SELECT PRIMARY gegeven. Daarna het commando JOIN. Hiermee wordt dBASE op het eerste record van het primaire USE-bestand geplaatst en de FOR-uitdrukking geëvalueerd voor elk record in het secundaire USE-bestand. Telkens wanneer de uitdrukking een TRUE (waar) oplevert, wordt aan de nieuwe database een record toegevoegd. Wanneer het einde van het secundaire USE-bestand wordt bereikt gaat het primaire USE-bestand één record vooruit en wordt het secundaire USE-bestand "teruggespoeld". Deze procedure gaat door tot het einde van het primaire USE-bestand.

Als de term FIELDS wordt weggelaten, bestaat de output-database uit alle velden van het primaire USE-bestand plus zoveel velden van het secundaire USE-bestand, als de dBASE-limiet van 32 velden toelaat.

Als de term FIELDS wordt opgegeven, worden alleen de velden die in de veldlijst staan in de output-database geplaatst.

Als de betrokken databases groot zijn, duurt dit commando erg lang. En als het samenvoegingscriterium niet erg beperkend is, wat veel samenvoegingen per primair record tot gevolg heeft, dan bestaat het gevaar dat dBASE de JOIN niet kan afwerken. Bijvoorbeeld: stel dat het primaire en het secundaire USE-bestand elk 1000 records bevatten en dat de <uitdrukking> altijd waar is, dan zou JOIN een database met één miljoen records moeten opbouwen, hetgeen onmogelijk is, aangezien het dBASE-maximum 65.535 records is.

Voorbeeld:

```
.USE VOORRAAD
```

```
.DISPLAY STRUCTURE
```

```
STRUCTURE FOR FILE:      VOORRAAD.DBF
NUMBER OF RECORDS:      00008
DATE OF LAST UPDATE:    15/12/83
PRIMARY USE DATABASE
```

FLD	NAME	TYPE	WIDTH	DEC
001	ITEM	C	020	
002	PRIJS	N	010	002
003	ART:NR	C	005	
004	VOORRADIG	N	005	
TOTAL			00041	

.LIST				
00001	SCHAKELKLOK	9.99	24776	1
00002	AUTOMAAT (4A)	1.67	31415	18
00003	AUTOMAAT (16A)	16.33	92653	7
00004	NUBE-DRAAIBANK	134999.00	89793	3
00005	KRUK, BAR	34.72	21828	77
00006	STOEL, BRUIN	198.37	76767	76
00007	STOEL, GROEN	200.00	70296	5
00008	KOPERDRAAD	22.00	11528	16

.SELECT SECONDARY

.USE ORDERS

.DISPLAY STRUCTURE

STRUCTURE FOR FILE: ORDERS.DBF
 NUMBER OF RECORDS: 00008
 DATE OF LAST UPDATE: 20/01/84
 PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
001	KLANT	C	020	
002	ART:NR	C	005	
003	AANTAL	N	005	
TOTAL			00031	

.LIST				
00001	JANSSEN, JOOP	31415	13	
00002	JANSSEN, JOOP	76767	13	
00003	HARRIS, KAREL	11528	44	
00004	ADAMS, JAN	89793	12	
00005	MACK, JAY	31415	3	
00006	DERVING, PIETER	76767	5	
00007	JUAN, DON	21828	5	
00008	GON, POLLIE	70296	9	

.SELECT PRIMARY

JOIN TO ORDLIJST FOR ART:NR = S.ART:NR;
 FIELD KLANT,ITEM,AANTAL,PRIJS

Gebruik voorraad-
 bestand om artikel-
 namen toe te voegen
 aan de orders.

92 ... JOIN

.USE ORDLIJST

.DISPLAY STRUCTURE

STRUCTURE FOR FILE: ORDLIJST.DBF
NUMBER OF RECORDS: 00008
DATE OF LAST UPDATE: 20/01/84
PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
001	KLANT	C	020	
002	ITEM	C	020	
003	AANTAL	N	005	
004	PRIJS	N	010	002
TOTAL			00056	

.LIST

00001	JANSSEN, JOOP	AUTOMAAT (4A)	13	1.67
00002	MACK, JAY	AUTOMAAT (4A)	3	1.67
00003	ADAMS, JAN	NUBE-DRAAIBANK	12	134999.00
00004	JUAN, DON	KRUK, BAR	5	34.72
00005	JANSSEN, JOOP	STOEL, BRUIN	13	198.37
00006	DERVING, PIETER	STOEL, BRUIN	5	198.37
00007	GON, POLLIE	STOEL, GROEN	9	200.00
00008	HARRIS, KAREL	KOPERDRAAD	44	22.00

.USE VOORRAAD

(Voegt klantennamen en artikelen samen wanneer de voorraad niet volstaat om de orders te honoreren; bijvoorbeeld om de klanten hierover te informeren.)

.JOIN TO NABESTEL FOR ART:NR = S.ART:NR .AND. VOORRADIG;
< AANTAL FIELD KLANT,ITEM

.USE NABESTEL

.LIST

00001	ADAMS, JAN	NUBE-DRAAIBANK
00002	GON, POLLIE	STOEL, GROEN
00003	HARRIS, KAREL	KOPERDRAAD

LIST

- a. LIST [<bereik>] [FOR<uitdr>] [<uitdrijs>] [OFF]
[WHILE<uitdr>] [FIELDS<veldrijs>]
- b. LIST STRUCTURE
- c. LIST MEMORY
- d. LIST FILES [ON<disk drive>] [LIKE<masker>]
- e. LIST STATUS

LIST is hetzelfde als DISPLAY, alleen is daarbij de standaard-systeemwaarde voor bereik ALLE records en wacht WAIT niet om verder te gaan na een groep van 15 records. De commando's LIST STRUCTURE, LIST FILES en LIST MEMORY werken op precies dezelfde manier als het DISPLAY-commando.

Met SET DELETED ON geeft LIST alleen records weer die geen wismarkering gekregen hebben.

LOCATE

LOCATE [< bereik >] [FOR < uitdr >]
[CONTINUE]

Met dit commando wordt in het USE-bestand het eerste database-record opgezocht waarvoor de < uitdr > waar is. Als aan de uitdrukking voldaan wordt, verschijnt de volgende boodschap:

RECORD n

Het commando CONTINUE kan gebruikt worden om verder te zoeken. Tussen LOCATE en CONTINUE kunnen nog andere dBASE-commando's worden gegeven. In dat geval is het aantal tekens in FOR < uitdr > echter beperkt tot 128 i.p.v. tot 254. Zie CONTINUE.

Als het systeem de uitdrukking niet vindt, verschijnt de boodschap END OF FILE en blijft de database op het laatste record in het bestand staan. Als de NEXT-bepaling (zie bereik, hoofdstuk 9.1.) in dit commando gebruikt wordt en als het systeem de uitdrukking niet binnen het bereik van NEXT vindt, verschijnt de boodschap END OF LOCATE en blijft de database op het laatst geëvalueerde record staan.

Opmerking: Een LOCATE werkt sneller met een bestand dat in USE is zonder geïndexeerd bestand.

Voorbeelden:

.USE INKOPEN.

.LIST1

00001	BONEN # 303 BLIK	5	0.69
00002	BRODEN	2	0.89
00003	BIEFSTUK	4	3.59
00004	PAPIEREN BORDEN	1	0.79
00005	PLASTIC VORKEN	5	0.39
00006	GROENTE	2	0.49
00007	KAAS	1	1.79
00008	MELK (1/2 LIT.)	2	1.19
00009	HOUTSKOOL, 5 # ZAKKEN	2	0.69

.LOCATE FOR PRIJS > .70

RECORD: 00002

.CONTINUE

RECORD: 00003

.DISP ITEM

BIEFSTUK

.CONTINUE

RECORD: 00004

.CONTINUE

RECORD: 00007

.CONTINUE

RECORD: 00008

.CONTINUE

END OF FILE

LOOP

LOOP

Met dit commando worden binnen een DO WHILE alle er op volgende commando's overgeslagen en wordt de DO WHILE voorwaarde opnieuw geëvalueerd. LOOP wordt gebruikt om de DO WHILE-lussen in te korten; deze kunnen erg veel tijd in beslag nemen als ze lang zijn of commando's bevatten die overgeslagen kunnen worden. LOOP werkt structureel net als een ENDDO: het brengt u terug naar de DO WHILE die het nestingsniveau heeft ingeleid.

LOOP's zijn een verleidelijke, maar niet altijd even elegante oplossing om uit een DO WHILE-lus te stappen. Hieronder twee programma's die hetzelfde resultaat opleveren. De eerste versie gebruikt een LOOP, de tweede maakt gebruik van IF THEN ELSE. De tweede is qua structuur te verkiezen.

Voorbeeld 1

```
STORE 1 TO INDEX
DO WHILE INDEX < 10
  STORE INDEX + 1 TO INDEX
  IF ITEM = ' '
    SKIP
  LOOP
ENDIF
DO VERWERKING
ENDDO
```

Spring naar het volgende record als ITEM blanco is en keer terug naar het begin van de DO WHILE

Voorbeeld 2:

```
STORE 1 TO INDEX
DO WHILE INDEX < 10
  STORE INDEX + 1 TO INDEX
  IF ITEM = ' '
    SKIP
  ELSE
    DO VERWERKING
  ENDIF
ENDDO
```

MODIFY

- a. MODIFY STRUCTURE
- b. MODIFY COMMAND [<bestand>]

Met vorm a. van dit commando kan de gebruiker de structuur van een dBASE-bestand veranderen. Elke verandering is mogelijk. Velden kunnen worden toegevoegd of gewist en parameters van de velden (b.v. naam, type, lengte, nummer, aantal decimalen) kunnen worden gewijzigd.

MODIFY werkt in de database die in USE is. De bestaande structuur wordt op het scherm weergegeven, de wijzigingen worden als bij full-screen edit direct op het scherm aangebracht; twee controltoetsen werken hier echter anders: < control > -N last een blanco regel in op de plaats van de cursor en < control > -T wist de regel waar de cursor op staat. De andere control-toetsen werken zoals beschreven in hoofdstuk 9.

Opmerking: Het MODIFY STRUCTURE-commando wist ALLE gegevens-records die in het USE-bestand stonden vóór het MODIFY-commando gegeven werd. Om een structuur te wijzigen en de gegevens te behouden, moet de structuur eerst met het COPY-commando naar een werkbestand gekopieerd worden, waarna u het werkbestand in USE neemt. Daarna worden de veranderingen aangebracht en worden de oude gegevens met APPEND aan het werkbestand toegevoegd. De oorspronkelijke database en het werkbestand kunnen zonodig met RENAME hun oorspronkelijke naam terugkrijgen. Zie het voorbeeld op de volgende bladzijde.

Met vorm b. is een beperkte vorm van full-screen editing mogelijk in commandobestanden. Als <bestand> niet wordt opgegeven, dan wordt daar in een prompt naar gevraagd en indien dat bestand niet bestaat, wordt het alsnog gecreëerd. Nadat een commandobestand is gewijzigd, zal MODIFY de vorige versie van het bestand behouden met de uitbreidingsletters .BAK en de nieuwe versie opslaan met het type .CMD.

In MODIFY COMMAND werken de < control > -N en < control > -T wijzigingsfuncties zoals hierboven beschreven. < control > -Q breekt alle wijzigingen in het commandobestand af en negeert ze, < control > -W schrijft het gewijzigde bestand naar schijf en bewaart de vorige versie zoals hierboven beschreven.

Er zijn een aantal belangrijke beperkingen bij deze commandovorm:

- 1) regels kunnen slechts 77 tekens lang zijn (incl. < return > /regelsprong);
- 2) TAB-tekens worden omgezet in enkelvoudige spaties;
- 3) de cursor kan in een bestand slechts 4000 bytes 'achteruit' gaan;
- 4) in tegenstelling tot bepaalde tekstverwerkers is er geen mogelijkheid tot het zoeken van reeksen (search en search/replace functies) noch tot het verplaatsen of kopiëren van tekstblokken.

98 ... MODIFY

De full-screen cursortoetsen werken als bij het MODIFY COMMAND,
uitgezonderd:

- < control > -N last een blanco regel in op de plaats van de cursor
- < control > -T wist de regel waar de cursor op staat en brengt de regels daaronder omhoog
- < control > -W schrijft de wijzigingen die aan het bestand werden aangebracht, over op schijf en verlaat MODIFY COMMAND (< control > -O voor Superbrain)
- < control > -Q breekt alle wijzigingen af en negeert ze
- < control > -R rolt één regel omlaag
- < control > -C rolt één scherpagina omhoog

Voorbeeld:

```
.NOTE – VOORBEELD VAN HET WIJZIGEN VAN EEN STRUCTUUR  
.NOTE – ZONDER DE INFORMATIE IN HET BESTAND TE VERLIEZEN  
  
.USE VOORRAAD  
  
.COPY TO WERK  
  
.USE WERK  
  
.MODIFY STRUCTURE  
  
.APPEND FROM VOORRAAD  
  
.DELETE FILE VOORRAAD  
  
.USE  
  
.RENAME WORK TO VOORRAAD
```


NOTE

- a. NOTE [<tekenreeks>]
- b. * [<tekenreeks>]

Met dit commando kunnen opmerkingen in een commandobestand worden geplaatst. In tegenstelling tot het REMARK-commando, zendt NOTE de inhoud niet naar de outputeenheid.

Voorbeeld:

NOTE – laatste wijziging van dit programma op 4 juli 1983

* – wachtwoord drie keer fout ingevoerd: verlaat het programma

100 ... PACK

PACK

PACK

Met dit commando worden alle records verwijderd die met het DELETE-commando een wismarkering gekregen hebben. Als het PACK-commando eenmaal gegeven is, kunnen de gewiste records niet meer worden gerecupereerd.

Als het bestand waarin PACK records verwijderd geïndexeerd is en de indexbestanden in USE zijn, zorgt PACK ervoor dat de indexbestanden en het USE-bestand gelijktijdig worden bijgewerkt.

Naast PACK bestaat ook de mogelijkheid het oude bestand met COPY naar een nieuw bestand te kopiëren. Records met een wismarkering worden namelijk niet gekopieerd. Daarna kunt u het oude bestand wissen (of bewaren als reserve) en het nieuwe bestand een nieuwe naam geven.

De schijfruimte die CP/M voor een bestand gereserveerd heeft vermindert niet door het PACKen. Om de opslagruimte op de schijf terug te winnen, gebruikt u COPY TO <bestandsnaam> en wist u daarna het bronbestand. Dit is een beperking van het CP/M-besturingssysteem en niet van dBASE II.

Voorbeelden:

```
.USE B:INKOPEN
```

```
.LIST
```

00001	BONEN	5	0.75
00002	BRODEN	2	0.97
00003	BIEFSTUK	4	3.94
00004	PAPIEREN BORDEN	1	0.86
00005	PLASTIC VORKEN	5	0.42
00006	GROENTE	2	0.53
00007	KAAS	1	1.96
00008	MELK	2	1.30
00009	HOUTSKOOL	2	0.75

```
.DELETE RECORD 8
```

```
00001 DELETION(S)
```

```
.LIST
```

00001	BONEN	5	0.75
00002	BRODEN	2	0.97
00003	BIEFSTUK	4	3.94
00004	PAPIEREN BORDEN	1	0.86
00005	PLASTIC VORKEN	5	0.42
00006	GROENTE	2	0.53
00007	KAAS	1	1.96
00008	*MELK	2	1.30
00009	HOUTSKOOL	2	0.75

```
.PACK
PACK COMPLETE, 00008 RECORDS COPIED
```

```
.LIST
00001 BONEN           5          0.75
00002 BRODEN         2          0.97
00003 BIEFSTUK       4          3.94
00004 PAPIEREN BORDEN 1          0.86
00005 PLASTIC VORKEN  5          0.42
00006 GROENTE        2          0.53
00007 KAAS           1          1.96
00008 HOUTSKOOL      2          0.75
```

Een PACK is niet altijd noodzakelijk. Stel bijvoorbeeld dat een aantal records genegeerd moeten worden, maar dat deze wél in de database moeten blijven. Deze records zullen door COPY, door APPEND en door SORT gewoon genegeerd worden, COUNT daarentegen zal ze wél meetellen. Vaak moet men weten of een record een wismarkering heeft, om het al dan niet te verwerken. Het volgende voorbeeld is een deel van een commandobestand waarin een record met wismarkering wordt overgeslagen, waarna het programma bij het volgende record verder gaat.

```
DO WHILE .NOT. EOF
  LOCATE FOR NATURE = "TLM"
  IF .NOT.*
    .
    commando's
  .
  ENDIF
  CONTINUE
ENDO
```

Dit lange verhaal is eigenlijk niet nodig. Het commando SET DELETED ON zal er net zo goed voor zorgen dat alle records met een wismarkering worden overgeslagen.

QUIT

QUIT [TO < lijst commandobestanden >]

Met dit commando worden alle databasebestanden, commandobestanden en andere bestanden gesloten en komt de computer opnieuw onder CP/M besturing. De boodschap *****END RUN dBASE***** verschijnt.

Als de term TO erin gegeven wordt, worden door CP/M alle programma's in de < com bestandslijst > op volgorde uitgevoerd. Met deze mogelijkheid kunt u dBASE verlaten en direct andere software opstarten.

Het aantal programma's of CP/M-commando's dat kan worden uitgevoerd is onbeperkt, zolang de limiet van 254 tekens voor elk commando niet overschreden wordt. dBASE kan aan het einde van een commandoreeks opnieuw opgestart worden, maar dit hoeft niet. Zodra de commandoreeks is uitgevoerd komt de computer weer onder CP/M besturing. Onder MP/M is dit niet mogelijk.

Voorbeeld:

.QUIT TO 'DIR B:', 'PIP PRN: = ADRLIJST.TXT', 'DBASE CMDBEST'

In dit voorbeeld wordt dBASE verlaten, wordt de index van de B-drive opgevraagd, wordt PIP gestart om een bestand op de printer te kopiëren, en wordt tot slot dBASE opnieuw geladen met een commandobestand dat meteen de besturing zal overnemen.

READ**READ [NOUPDATE]**

Dit commando start de full-screen-modus voor het wijzigen en/of invoeren van variabelen die geïdentificeerd zijn met en weergegeven zijn door een "@"-commando met een GET-zin. De cursor kan naar elke GET-variabele gebracht worden. De invoer en wijzigingen die men op het scherm aanbrengt worden in de databasevelden of geheugenvariabelen opgeslagen.

Bij SET FORMAT TO (formatbestand) start READ de uitvoering van alle "@"-commando's in het formatbestand: het scherm wordt geformatteerd en alle GET-variabelen kunnen worden gewijzigd. Deze techniek is een eenvoudig alternatief voor het EDIT-commando wanneer men in de interactieve modus werkt.

Bij SET FORMAT TO SCREEN gebruikt men een ERASE-commando om het scherm te wissen. Vervolgens kan met een reeks "@"-commando's het scherm worden geformatteerd. Om te kunnen wijzigen en invoeren is vervolgens een READ-commando vereist.

Als na een READ-commando een tweede of latere reeks "@"-commando's gegeven wordt, plaatst READ de cursor op de eerste GET-variabele na de laatste READ. Op deze manier kunnen het schermformat en de bijzondere gewijzigde variabelen door de gebruiker bepaald worden via zijn antwoord op vorige READ-commando's.

Variabelen, die met de "@"-commando's gebruikt worden en met het READ-commando zijn gewijzigd, moeten als veldnamen in het USE-bestand staan of geheugenvariabelen van het type tekenreeks zijn. Geheugenvariabelen moeten elders gedefinieerd worden voordat het "@"-commando gegeven wordt. Om een geheugenvariabele te initialiseren, kunt u deze eerst STOREn met zoveel spaties als de maximum lengte van de variabele (bijvoorbeeld: STORE ' ' to GEHVAR).

Zie hoofdstuk 8 voor cursorbesturing en gegevensinvoer.

SET SCREEN ON moet in werking zijn (dit is de standaard-systeemwaarde die bij de installatie van dBASE voor full-screen bewerking is gekozen).

De optie NOUPDATE schakelt het bijwerken (updaten) van indexbestanden uit. Na een READ-commando controleert dBASE of er indexsleutels zijn gewijzigd: de indexbestanden moeten in dat geval immers bijgewerkt worden. Bij meervoudige indexen kan dat bijwerken enige tijd vergen. Als u zeker weet dat het READ-commando géén sleutels gaat wijzigen (maar alleen geheugenvariabelen of alleen velden die geen sleutel vormen), dan gebruikt men READ NOUPDATE, zodat dBASE deze sleutel-controle overslaat.

104 ... READ

Voorbeeld:

```
.  
. .  
STORE ' ' TO BWIJZE  
STORE ' ' TO KNR  
ERASE  
@5,0 SAY 'Voer C in voor contante betaling'  
@6,0 SAY ' of F voor betaling na factuur'  
@8,5 GET BWIJZE  
READ  
IF BWIJZE = 'F'  
  @10,10 SAY 'Voer klantnummer in' GET KNR PICTURE 'KL/9999'  
  READ  
ENDIF  
. .  
.
```

In dit deel van een commandobestand wordt het scherm geschoond en worden de eerste twee "@"-commando's gegeven. De cursor staat tussen twee dubbele punten die de schermpositie van de variabele BWIJZE aangeven. Omdat met de eerste STORE de lengte van BWIJZE op 1 teken gesteld is, wordt BWIJZE door één aanslag van de gebruiker gevuld en wordt het eerste READ-commando verlaten.

Als de gebruiker een "F" heeft ingevoerd, wordt het "@"-commando uitgevoerd waarin naar het klantnummer gevraagd wordt. U ziet dat KNR in de STORE lang genoeg gedefinieerd is zodat er voldoende plaats is voor de constante "KL/" die de PICTURE-bepaling in de "@" opgeeft.

Voorbeeld:

```

USE CHEQUES
SET FORMAT TO SCREEN
ACCEPT "Keuze" TO KEUZE
IF KEUZE$'Aa'
  ERASE
  DO WHILE NUMMER < > 0
    APPEND BLANK
    @ 5,0 SAY "Voer volgend nummer in";
    GET NUMMER PICTURE '99999'
    @ 6,0 SAY "Voer begunstigde in";
    GET BEGUNST PICTURE;
    'XXXXXXXXXXXXXXXXXXXXXXXXXXXX'
    @ 7,0 SAY "Voer bedrag in";
    GET BEDRAG PICTURE '9999999999'
    @ 8,5 SAY "Voer beschrijving aankoop in";
    GET BESCHR
    @ 8,40 SAY "Bankdebet al ontvangen?";
    GET BDEB
  READ
  ENDDO
ENDIF

```

In het laatste voorbeeld werd een bestand gebruikt en rechtstreeks gewijzigd. Om het programma te verlaten kan de gebruiker als chequenummer nul invoeren.

Zie het "@"-commando voor bijzonderheden.

106 ... RECALL

RECALL

```
RECALL [<bereik>][FOR <uitdr>]
      [WHILE <uitdr>]
```

Dit commando verwijdert wismarkeringen die met DELETE werden aangebracht.

Voorbeelden:

```
.USE DUPE3
```

```
.LIST
```

```
00001  TIMMERMANS, THEO          1357
00002  KNOOPS, HERMAN           2468
00003  BERINGS, MARIJKE        3344
00004  SMULDERS, KEES          6743
00005  PAPAMOUROLOS, D.       1011
00006  VAN DE HEUVEL, WILLIAM  6623
```

```
.3
```

```
.DELETE NEXT 3
00003 DELETION(S)
```

```
.LIST
```

```
00001  TIMMERMANS, THEO          1357
00002  KNOOPS, HERMAN           2468
00003  *BERINGS, MARIJKE        3344
00004  *SMULDERS, KEES          6743
00005  *PAPAMOUROLOS, D.       1011
00006  VAN DE HEUVEL, WILLIAM  6623
```

```
.RECALL RECORD 4
00001 RECALL(S)
```

```
.LIST
```

```
00001  TIMMERMANS, THEO          1357
00002  KNOOPS, HERMAN           2468
00003  *BERINGS, MARIJKE        3344
00004  SMULDERS, KEES          6743
00005  *PAPAMOUROLOS, D.       1011
00006  VAN DE HEUVEL, WILLIAM  6623
```


.RECALL ALL
00002 RECALL(S)

.LIST
00001 TIMMERMANS, THEO 1357
00002 KNOOPS, HERMAN 2468
00003 BERINGS, MARIJKE 3344
00004 SMULDERS, KEES 6743
00005 PAPAMOUROLOS, D. 1011
00006 VAN DE HEUVEL, WILLIAM 6623

108 ... REINDEX

REINDEX

REINDEX

Dit commando werkt zoals het commando INDEX ON <uitdr> TO <indexbestandsnaam>, alleen hoeft de sleuteluitdrukking en het TO-bestand niet te worden opgegeven. REINDEX gebruikt de sleutels uit de indexbestanden die in USE zijn en bouwt de bestanden helemaal opnieuw op.

.USE BOEKEN INDEX AUTEURS

.REINDEX

RELEASE

```
RELEASE [<geheugenvarijst>]
      [ALL]
      [ALL LIKE <masker>]
      [ALL EXCEPT <masker>]
```

Dit commando maakt alle of bepaalde geheugenvariabelen vrij zodat de ingenomen ruimte weer beschikbaar komt voor nieuwe geheugenvariabelen. Als ALL gespecificeerd wordt, worden alle geheugenvariabelen gewist.

Het <masker> is ongeveer gelijk aan <masker> in DISPLAY FILES. Vraagtekens maskeren een bepaald teken en sterretjes alle volgende tekens.

Voorbeeld: RELEASE ALL LIKE MEM??X zal de geheugenvariabelen MEMABX, MEMFFX en MEMRRX vrijmaken, maar MEMAB, MEMABY of MAMABX niet.

Met RELEASE ALL EXCEPT MEM* worden MAM, MAMBX, en ME vrijgemaakt, maar MEM, MEMXXXXX of MEMABX niet.

110 ... REMARK

REMARK

REMARK [tekenreeks]

Dit commando kan een willekeurige reeks tekens weergeven. De inhoud van dit commando wordt op de actieve output-eenheid weergegeven.

Voorbeelden:

```
.REMARK *****DIT IS EEN TEST*****  
*****DIT IS EEN TEST*****
```

RENAME

RENAME <oorspronkelijke bestandsnaam> TO <nieuwe bestandsnaam>

Met dit commando kan de naam van het bestand in de CP/M-index worden gewijzigd. Als er geen bestandstype wordt gegeven (max. 3 tekens na de bestandsnaam), gaat dBASE ervan uit dat de naam van een database gebruikt wordt, en wordt .DBF het bestandstype voor genoemde bestanden. Zie hoofdstuk 4 voor meer informatie over bestandstypen in dBASE.

Voorbeeld:

.RENAME VOORACTU TO VOOR1982

.RENAME D:RAPPORT.FRM TO RAPPORT.BAK

.RENAME Z:TYPE TO MET:TYPE.TYP

REPLACE

```

REPLACE [<bereik>] <veld> WITH <uitdr>
      [, <veld> WITH <uitdr>]
      [FOR <uitdr>] [NOUPDATE]
      [WHILE <uitdr>]

```

Met dit commando wordt de inhoud van gespecificeerde gegevensvelden van het bestand in USE vervangen door nieuwe gegevens. In tegenstelling tot het STORE-commando wijzigt REPLACE alleen veldvariabelen terwijl STORE alleen geheugenvariabelen wijzigt.

Als <bereik> niet in het commando gegeven wordt, werkt REPLACE alleen op het lopende record.

Als REPLACE een indexsleutel wijzigt en de index in USE is, wordt het indexbestand aangepast: de oude indexinvoer wordt gewist en de nieuwe wordt op de juiste plaats ingelast. Indexbestanden die niet in USE zijn worden niet aangepast. Als een REPLACE-commando een indexsleutel wijzigt verschuift het gewijzigde record binnen het bestand; het nieuwe "volgende record" kan een heel ander zijn dan het oude "volgende record". Sleutels REPLACEn met NEXT n als <bereik> kan geheel onvoorspelbare gevolgen hebben en is beslist af te raden.

De optie NOUPDATE schakelt het bijwerken (updaten) van indexbestanden uit. Na een REPLACE-commando controleert dBASE of er indexsleutels werden gewijzigd: de indexbestanden moeten in dat geval immers bijgewerkt worden. Bij meervoudige indexen kan die bijwerking enige tijd duren. Als u zeker weet dat het REPLACE-commando géén sleutels gaat wijzigen (maar alleen velden die geen sleutel vormen), dan gebruikt u NOUPDATE, zodat dBASE de indexaanpassing overslaat.

Opmerking: Als u PRIMARY en SECONDARY databases gebruikt, kan een veld *alleen* veranderd worden als het in de huidige geSELECTeerde database staat.

Voorbeelden:

```
.USE INKOPEN
```

```
.NOTE INFLATIE GEEFT 10% PRIJSVERLAGING
```

```
.LIST
```

00001	BONEN # 303 BLIK	5	0.69
00002	BRODEN	2	0.89
00003	BIEFSTUK	4	3.59
00004	PAPIEREN BORDEN	1	0.79
00005	PLASTIC VORKEN	5	0.39

REPLACE ... 113

00006	GROENTE	2	0.49
00007	KAAS	1	1.79
00008	MELK (1/2 LIT.)	2	1.19
00009	HOUTSKOOL, 5# ZAKKEN	2	0.69

.REPLACE ALL PRIJS WITH PRIJS*1.1

00009 REPLACEMENT(S)

.LIST

00001	BONEN # 303 BLIK	5	0.75
00002	BRODEN	2	0.97
00003	BIEFSTUK	4	3.94
00004	PAPIEREN BORDEN	1	0.86
00005	PLASTIC VORKEN	5	0.42
00006	GROENTE	2	0.53
00007	KAAS	1	1.96
00008	MELK (1/2 LIT.)	2	1.30
00009	HOUTSKOOL, 5# ZAKKEN	2	0.75

USE B:INKOPEN

.COPY TO B:INKOP2

00009 RECORDS COPIED

.LIST

00001	BONEN	5	0.75
00002	BRODEN	2	0.97
00003	BIEFSTUK	4	3.94
00004	PAPIEREN BORDEN	1	0.86
00005	PLASTIC VORKEN	5	0.42
00006	GROENTE	2	0.53
00007	KAAS	1	1.96
00008	MELK	2	1.30
00009	HOUTSKOOL	2	0.75

.GOTO TOP

.REPLACE NEXT 5 PRIJS WITH PRIJS*1.1 FOR PRIJS > .75

00003 REPLACEMENT(S)

.LIST

00001	BONEN	5	0.75
00002	BRODEN	2	1.06
00003	BIEFSTUK	4	4.33
00004	PAPIEREN BORDEN	1	0.94
00005	PLASTIC VORKEN	5	0.42
00006	GROENTE	2	0.53
00007	KAAS	1	1.96
00008	MELK	2	1.30
00009	HOUTSKOOL	2	0.75

114 ... REPLACE

.USE CHEQUES

.DISP STRU

STRUCTURE FOR FILE: CHEQUES.DBF
NUMBER OF RECORDS: 00016
DATE OF LAST UPDATE: 18/10/82
PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
001	NUMMER	N	005	
002	BEGUNST	C	020	
003	BEDRAG	N	010	002
004	INKOMEND	L	001	
005	UITGAAND	L	001	
TOTAL			00038	

.LIST

00001	1	Telefoon	104.89	.F.	.T.
00002	2	Gas	4.15	.F.	.T.
00003	3	Elektriciteit	250.30	.F.	.T.
00004	4	Supermarkt	1034.45	.F.	.T.
00005	134	Ik, loon	561.77	.T.	.F.
00006	6	Bank	4.00	.T.	.T.
00007	7	Dokter De Groot	100.00	.T.	.T.
00008	8	PC Shop	100.00	.F.	.T.
00009	9	Reparatie auto	500.01	.F.	.T.
00010	10	Ik	561.77	.T.	.F.
00011	11	Tupperware	50.02	.F.	.T.
00012	12	Ik	561.77	.T.	.F.
00013	13	Ik	750.03	.T.	.F.
00014	234	Peter Jansen	14.00	.F.	.T.
00015	237	Restaurant Hapklaar	650.00	.F.	.T.
00016	30	Ik	561.77	.T.	.F.

.11

.REPLACE INKOMEND WITH F
00001 REPLACEMENT(S)

.DISPLAY

00011	11	Tupperware	50.02	.F.	.T.
-------	----	------------	-------	-----	-----

REPORT

REPORT [FORM <formulierbestand >] [<bereik >] [TO PRINT] [PLAIN]
 [FOR <uitdr >]
 [WHILE <uitdr >]

REPORT wordt gebruikt om rapporten te maken (op scherm of printer) waarin de gegevens uit het bestand in USE op een bepaalde manier worden weergegeven. Rapporten kunnen bestaan uit kolommen met titels, getotaliseerde numerieke velden, uitdrukkingen m.b.t. gegevensvelden, geheugenvariabelen en constanten.

Met de FOR-bepaling wordt alleen die informatie gerapporteerd, die aan de voorwaarden van de <uitdr > voldoet. De TO PRINT-bepaling stuurt het rapport naar de printer en naar het scherm. Tenzij u iets anders opgeeft, neemt het systeem ALL als waarde voor het <bereik >.

Wanneer het REPORT-commando voor de eerste keer gebruikt wordt (voor een nieuw rapport) wordt een formulierbestand opgebouwd. dBASE vraagt de gebruiker om specificaties over het rapportformat en genereert automatisch het formulierbestand. Dit formulierbestand kan ook voor latere rapporten gebruikt worden zodat men de specificaties niet opnieuw hoeft op te geven. Als u niet met behulp van het FORM-deel een naam opgeeft voor het formulierbestand, wordt daar in een prompt naar gevraagd.

In het volgende voorbeeld van een formulierbestand worden bijna alle opties gespecificeerd. Met de optie "M" kan de gebruiker het aantal spaties bepalen voor de linker marge van het rapport (de systeemwaarde is 8 spaties); met de optie "L" wordt het aantal regels per pagina veranderd (de systeemwaarde is 57 regels); en met de optie "W" wordt de plaats van het pagina-opschrift geregeld (systeemwaarde voor de paginabreedte is 80 tekens) aangezien "W" alleen voor het centreren van de kop gebruikt wordt.

Het REPORT-commando verwerkt maximaal 24 velden.

Voorbeeld:

```
.REPORT FORM INKOPFRM
ENTER OPTIONS? M= LEFT MARGIN, L= LINES/PAGE, W= PAGE
WIDTH M= 5, W= 65
PAGE HEADING? (Y/N)Y
ENTER PAGE HEADING: Boodschappenlijst voor picknick
DOUBLE SPACE REPORT? (Y/N)N
ARE TOTALS REQUIRED? (Y/N)Y
SUBTOTALS IN REPORT? (Y/N)N
COL      WIDTH,CONTENTS
001      23,ITEM + '...'
ENTER HEADING: Item; = = = =
002      10,AANTAL
ENTER HEADING: > Aantal; = = = = =
ARE TOTALS REQUIRED? (Y/N)Y
003      10,PRIJS
ENTER HEADING: > Prijs/Item; = = = = =
ARE TOTALS REQUIRED? (Y/N)N
004      10,AANTAL*PRIJS
ENTER HEADING: > Prijs; = = = =
ARE TOTALS REQUIRED? (Y/N)Y
005      <return >
```

REPORT vraagt naar de breedte en de inhoud van elk veld dat geprint moet worden. De breedte die hier gevraagd wordt, is niet hetzelfde als de feitelijke breedte van het te printen veld; in de eerste kolom van het hierboven gegeven rapportformulier bijvoorbeeld, is de ITEM-kolom 23 tekens breed, terwijl deze kolom in de database slechts 20 tekens breed is. Deze 3 extra tekens in het rapport zijn nodig voor de reeks '...' die aan de inhoud van het veld ITEM wordt toegevoegd. Dit betekent ook dat als de rapportkolom smaller is dan het veld dat in de kolom moet komen, dBASE het veld over meerdere regels zal spreiden zodat het in de kolom past. Een veld van 80 tekens dat in een kolom van 50 tekens geplaatst wordt, zal 2 regels innemen.

De inhoud van de kolommen kunnen velden uit een database, een geheugenvariabele, konstanten of uitdrukkingen zijn. Merk op dat in kolom 1 van het formulier van de vorige bladzijde een samengevoegde reeks komt. *In het rapport* krijgt elk record uit de database in USE drie puntjes toegevoegd aan het eind van de reeks (*het bestand zelf blijft ongewijzigd*). Kolom vier bevat AANTAL en PRIJS. Deze kolom heeft geen veldequivalent in de database. (De velden zijn, van links naar rechts, ITEM, AANTAL en PRIJS).

.LIST			
00001	BONEN	5	0.75
00002	BRODEN	2	1.06
00003	BIEFSTUK	4	4.33
00004	PAPIEREN BORDEN	1	0.94
00005	PLASTIC VORKEN	5	0.42
00006	GROENTE	2	0.53
00007	KAAS	1	1.96
00008	MELK	2	1.30
00009	HOUTSKOOL	2	0.75

Voor het FORM-bestand (de specificaties over het rapport) wordt in de titels een aantal speciale tekens gebruikt. Voor pagina- en kolomtitels en tekenreeksen wordt een punt-komma (;) gebruikt om het opschrift bij de punt-komma af te breken en de weergave op de volgende regel verder te laten gaan. Als een opschrift of reeks te lang is voor het toegestane aantal spaties, dan wordt deze (indien mogelijk) afgebroken bij de laatste spatie en op de volgende regel afgemaakt. De andere significante tekens zijn "<", en ">". Als voor een kolomopschrift een "<" staat dan wordt het opschrift links aangesloten. Bij ">" wordt het rechts aangesloten.

Andere opties in REPORT zijn totaliseren, subtotaliseren en het maken van samenvattende rapporten. In samenvattende rapporten worden de records niet weergegeven, maar alleen de totalen en subtotalen. Totalen en subtotalen worden alleen voor numerieke velden berekend (zie voorbeelden).

Een <return> beëindigt het rapportformulier en start de weergave. Als de zinsnede TO PRINT in het eerste commando gegeven werd, wordt op de printer een kopie van het rapport afgedrukt.

Andere dBASE-commando's voor de rapportbewerking zijn "SET EJECT OFF", "SET HEADING TO" en "SET DATE TO". Voordat REPORT de informatie uitprint, wordt eerst een nieuwe pagina gegeven. Deze functie kan worden opgeheven met het SET EJECT OFF-commando. Met het SET HEADING TO-commando kan het rapport tijdens de uitvoering van een extra titel worden voorzien. Dit commando werkt gedurende één sessie. (De titel moet bij elk begin van dBASE-sessie opnieuw bepaald worden.) Hetzelfde geldt voor het SET DATE TO-commando. De datum van het rapport kan met dit commando gewijzigd of weggelaten worden. Zie het SET-commando voor meer informatie.

REPORT biedt vele mogelijkheden, maar er zullen zeker toepassingen zijn waarvoor deze functie niet volstaat (bijzondere formulieren, meer complexe manier om gegevens op te vragen). In die gevallen zijn "@" en SET FORMAT TO PRINT de aangewezen commando's. Zie "@"-commando voor meer informatie en voorbeelden.

Voorbeeld 1:

.USE INKOPEN

.REPORT FORM INKOPFRM

PAGE NO. 00001

Boodschappenlijst voor picknick

Item	Aantal	Prijs/Item	Prijs	
BONEN	...	5	0.75	3.75
BRODEN	...	2	1.06	2.12
BIEFSTUK	...	4	4.33	17.32
PAPIEREN BORDEN	...	1	0.94	0.94
PLASTIC VORKEN	...	5	0.42	2.10
GROENTE	...	2	0.53	1.06
KAAS	...	1	1.96	1.96
MELK	...	2	1.30	2.60
HOUTSKOOL	...	2	0.75	1.50
TOTAL				
	24			33.35

.SET HEADING TO 4 juli 1976

.REPORT FORM INKOPFRM

PAGE NO. 00001

4 juli 1976

Boodschappenlijst voor picknick

Item	Aantal	Prijs/Item	Prijs	
BONEN	...	5	0.75	3.75
BRODEN	...	2	1.06	2.12
BIEFSTUK	...	4	4.33	17.32
PAPIEREN BORDEN	...	1	0.94	0.94
PLASTIC VORKEN	...	5	0.42	2.10
GROENTE	...	2	0.53	1.06
KAAS	...	1	1.96	1.96
MELK	...	2	1.30	2.60
HOUTSKOOL	...	2	0.75	1.50
TOTAL				
	24			33.35

Voorbeeld 2:

Dit voorbeeld illustreert het subtotaliseren met dBASE. Als het rapportformulier gecreëerd wordt, worden de subtotalen berekend voor het veld ART:NR. Zo weet men niet alleen wie het artikel heeft besteld, maar ook hoeveel stuks van elk artikel geproduceerd (of ingekocht) moeten worden.

```
.USE ORDERS INDEX ORDERS
```

```
.LIST
00003 HARRIS, ARNOLD      11528      44
00013 ANDERSON, JAMES    11528      16
00007 JUAN, DON          21828      5
00001 SWARTZ, JOE        31415     13
00005 MACK, JAY          31415      3
00009 BARNETT, WALT     31415      6
00008 SALT, CLARA       70296      9
00002 SWARTZ, JOE        76767     13
00006 TERRY, HANS       76767      5
00010 NICHOLS, BILL     76767     17
00004 ADAMS, JEAN       89793     12
00011 MURRAY, CAROL     89793      4
00012 WARD, CHARLES A.  92653     15
```

```
.REPORT
```

```
ENTER REPORT FORM NAME:ORDERS
```

```
ENTER OPTIONS, M = LEFT MARGIN, L = LINES/PAGE, W = PAGE
```

```
WIDTH W = 65
```

```
PAGE HEADING? (Y/N)Y
```

```
ENTER PAGE HEADING: BESTELLINGEN PER ARTIKELNUMMER
```

```
DOUBLE SPACE REPORT, (Y/N)N
```

```
ARE TOTALS REQUIRED? (Y/N)Y
```

```
SUBTOTALS IN REPORT? (Y/N)Y
```

```
ENTER SUBTOTALS FIELD: ART:NR
```

```
SUMMARY REPORT ONLY? (Y/N)N
```

```
EJECT PAGE AFTER SUBTOTALS? (Y/N)N
```

```
ENTER SUBTOTAL HEADING: Bestellingen voor artikelnummer
```

```
COL      WIDTH,CONTENTS
```

```
001      20,KLANT
```

```
ENTER HEADING: < NAAM KLANT
```

```
002      10,AANTAL
```

```
ENTER HEADING: > BESTELDE HOEVEELHEID
```

```
ARE TOTALS REQUIRED? (Y/N)Y
```

```
003
```

PAGE NO. 00001

BESTELLINGEN PER ARTIKELNUMMER

NAAM KLANT	BESTELDE HOEVEELHEID
Bestellingen voor artikelnummer 11528	
HARRIS, ARNOLD	44
ANDERSON, JAMES	16
** SUBTOTAL **	60
*Bestellingen voor artikelnummer 21828	
JUAN, DON	5
** SUBTOTAL **	5
*Bestellingen voor artikelnummer 31415	
SWARTZ, JOE	13
MACK, JAY	3
BARNETT, WALT	6
** SUBTOTAL **	22
*Bestellingen voor artikelnummer 70296	
SALT, CLARA	9
** SUBTOTAL **	9
*Bestellingen voor artikelnummer 76767	
SWARTZ, JOE	13
TERRY, HANS	5
NICHOLS, BILL	17
** SUBTOTAL **	35
*Bestellingen voor artikelnummer 89793	
ADAMS, JEAN	12
MURRAY, CAROL	4
** SUBTOTAL **	16
*Bestellingen voor artikelnummer 92653	
WARD, CHARLES A.	15
** SUBTOTAL **	15
** TOTAL **	162

Voorbeeld 3:

Stel dat u met uw collega's gaat kaarten en dat de verliezer een etentje betaalt. Om eerlijk te spelen houdt u de scores bij. Uit de database kunnen allerlei gegevens gehaald worden (zoals wie er gaat verliezen als hij niet oplet). De volgende database is een voorbeeld van zo'n spelletje.

.DISP STRU

STRUCTURE FOR FILE: KAARTEN.DBF
 NUMBER OF RECORDS: 00016
 DATE OF LAST UPDATE: 17/09/83
 PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
001	DATUM	C	008	
002	LISA	N	003	
003	ANNA	N	003	
004	WILLEM	N	003	
TOTAL			00018	

.REPORT

ENTER REPORT FORM NAME: KAARTEN
 ENTER OPTIONS, M = LEFT MARGIN, L = LINES/PAGE, W = PAGE
 WIDTH W = 40
 PAGE HEADING? (Y/N)Y
 ENTER PAGE HEADING: Scores Whist
 DOUBLE SPACE REPORT, (Y/N)N
 ARE TOTALS REQUIRED? (Y/N)Y
 SUBTOTALS IN REPORT? (Y/N)N
 COL WIDTH, CONTENTS
 001 10, DATUM
 ENTER HEADING: Datum; Spel
 002 6, Lisa
 ENTER HEADING: Score; Lisa
 ARE TOTALS REQUIRED? (Y/N)Y
 003 6, ANNA
 ENTER HEADING: Score; Anna
 ARE TOTALS REQUIRED? (Y/N)Y
 004 6, WILLEM
 ENTER HEADING: Score; Willem
 ARE TOTALS REQUIRED? (Y/N)Y
 005 5, LISA + ANNA + WILLEM
 ENTER HEADING: Spel; Totaal
 ARE TOTALS REQUIRED? (Y/N)Y
 006 <return>

(Opmerking – De laatste kolom in het rapportformulier is een totaal van de

122 ... REPORT

scores in elk record, dus de som voor Lisa, Willem en Anna. Een kolom hoeft niet in de database te bestaan om gebruikt te kunnen worden; het veld "LISA + ANNA + WILLEM" bestaat niet in de database "KAARTEN". Dit voorbeeld laat zien hoe een uitdrukking in een rapport kan worden geplaatst.)

Page NO. 00001

Scores Whist				
Datum Spel	Score Lisa	Score Anna	Score Willem	Spel Totaal
26/05/83	29	75	53	157
27/05/83	45	48	63	156
28/05/83	50	56	74	180
29/05/83	86	24	72	182
05/06/83	43	12	75	130
12/06/83	42	9	27	78
26/06/83	84	35	63	182
06/07/83	33	71	26	130
19/08/83	37	55	38	130
15/09/83	19	57	54	130
16/09/83	15	7	108	130
17/09/83	59	13	58	130
** TOTAL **	542	462	711	1715

Een rapport kan ook slechts enkele records uit een bestand bevatten:

.GOTO RECORD 7

.REPORT NEXT 4 FORM KAARTEN

PAGE NO. 00001

Scores Whist				
Datum Spel	Score Lisa	Score Anna	Score Willem	Spel Totaal
07/07/83	40	63	27	130
09/07/83	55	41	60	156
13/07/83	40	63	54	157
23/07/83	38	69	23	130
** TOTAL **	173	236	164	573

In een rapport kan ook informatie staan die aan bepaalde criteria voldoet:

.REPORT FORM KAARTEN FOR WILLEM < 50

PAGE NO. 00001

Scores Whist

Datum Spel	Score Lisa	Score Anna	Score Willem	Spel Totaal
12/06/83	42	9	27	78
06/07/83	33	71	26	130
07/07/83	40	63	27	130
23/07/83	38	69	23	130
19/08/83	37	55	38	130
** TOTAL **	190	267	141	598

.REPORT FORM ORDLST WHILE KLANT > = "M"

PAGE NO. 00001

13/12/83

KLANT	ARTIKEL	AANTAL
MACK, JAY	31415	3
MURRAY, CAROL	89793	4
NICHOLS, BILL	76767	17
SALT, CLARA	70296	9
SWARTZ, JOE	31415	15
SWARTZ, JOE	76767	13
TERRY, HAS	76767	5
WARD, CHARLES A.	92653	15

PLAIN is een uitbreiding van het REPORT-commando. Hiermee kan een dBASE-rapport op zodanige manier gecreëerd worden dat het ingelast kan worden in een rapport dat door een tekstverwerker is opgesteld.

Met de PLAIN-bepaling worden paginanummers en de datum bovenaan de rapportpagina weggelaten. Pagina-titels worden alleen aan het begin van het dBASE-rapport ingelast. Als u geen pagina-opvoer tussen de verschillende rapporten wilt, gebruikt u het SET EJECT OFF.

Voorbeelden:

.USE ONTWERP INDEX DOC

.NOTE POINTER OP EERSTE RECORD VOOR RAPPORT ZETTEN

.304

.REPORT FORM TABELS PLAIN WHILE DOC = "3 - 280 - T"

ENTER OPTIONS, M = LEFT MARGIN, L = LINES/PAGE, W = PAGE WIDTH

PAGE HEADING? (Y/N)Y

ENTER PAGE HEADING: TABELLEN

DOUBLE SPACE REPORT? (Y/N)N

ARE TOTALS REQUIRED? (Y/N)N

COL WIDTH, CONTENTS

001 20, \$(DOC, 7, 17)

ENTER HEADING: TABELLEN

002 40, BESCHRIJV

ENTER HEADING: BESCHRIJVING

003 <return>

TABELLEN

TABEL	BESCHRIJVING
Tabel 1	GLL telemetrische modes
Tabel 2	Toegestane combinaties van R/T en Record-formats
Tabel 2.3.2	Codes Gebruikersbus
Tabel 3	GLL Bitsnelheids-toekenning
Tabel 4	Format header
Tabel 5	Vaststelling format
Tabel 6	Voortgangscontrole
Tabel 7	S/C Klok-vooruitgang
Tabel A2.2.1	Ing gegevens layout
Tabel A2.2.2	Identificatie vaste-zone-structuur/plaats
Tabel A2.2.3	Identificatie variabele zone-zakstructuur/plaats
Tabel A2.2.4	Voorbeeld-tijd meting CDS-vaste zone
Tabel A2.2.8	Engr-metingen

RESET

RESET [<drive>]

Nadat een diskette verwisseld is, wordt CP/M met het RESET-commando opnieuw ingesteld. Als een diskette verwisseld wordt kan men gewoonlijk niets invoeren zonder een warme of zachte zelfstart (boot).

Als <drive> niet gespecificeerd wordt, dan wordt het gehele systeem opnieuw ingesteld. Jammer genoeg kan noch dBASE, noch het besturings-systeem bepalen welke diskettes u omwisselde, en er kunnen I/O-fouten ontstaan als een schijf met open bestanden verwisseld wordt. Als <drive> bepaald wordt, gaat dBASE na of er op die eenheid open bestanden staan, om I/O-fouten te voorkomen. Daarom zou RESET zonder de <drive>-bepaling eigenlijk nooit gebruikt mogen worden, maar de mogelijkheid is terwille van compatibiliteit toch opgenomen. De eenheid waarin de dBASE-systeem-commandobestanden zitten, mag niet verwisseld en geRESET worden.

Een RESET-commando zonder dat een schijf verwisseld wordt, heeft geen effect.

RESTORE

RESTORE FROM <bestand> [ADDITIVE]

Dit commando leest een bestand met geheugenvariabelen. Het bestand moet met het commando SAVE TO <bestand> zijn aangemaakt. Alle geheugenvariabelen aanwezig vóór het RESTORE-commando, worden door dit commando gewist.

Als de ADDITIVE wordt opgenomen, worden de eerder aanwezige geheugenvariabelen niet vrijgegeven en worden zoveel variabelen uit het FROM-bestand toegevoegd als mogelijk is.

Voorbeelden:

```
.DISPLAY MEMORY
EEN      (N)  1.0000
ALFABET  (C)  ABCDEFGHIJKL
TEKENS   (C)  ABCDEFGHIJKLMNOPQR ETC
**TOTAL**      03  VARIABLES USED      00042 BYTES USED

.SAVE TO MEMBEST

.RELEASE ALL

.DISPLAY MEMORY
** TOTAL **      00  VARIABLES USED      00000 BYTES USED

.RESTORE FROM MEMBEST

.DISPLAY MEMORY
EEN      (N)  1.0000
ALFABET  (C)  ABCDEFGHIJKL
TEKENS   (C)  ABCDEFGHIJKLMNOPQR ETC
** TOTAL **      03  VARIABLES USED      00042 BYTES USED
```

RETURN

RETURN

Dit commando wordt binnen een commandobestand gebruikt om de besturing over te brengen naar het commandobestand waarmee het werd opgeroepen (of naar het toetsenbord, als de gebruiker het commandobestand rechtstreeks heeft opgeroepen). Een End-Of-File aan het einde van een commandobestand heeft hetzelfde effect als een RETURN-commando.

Meestal staat er in een commandobestand een RETURN-commando op de laatste regel die uitgevoerd moet worden. De programmeur kan eronder dan tekst en commentaar schrijven die niet verwerkt worden.

SAVE

SAVE TO <bestand> [ALL LIKE <masker>]

Dit commando slaat alle huidige geheugenvariabelen op in een bestand. Met het RESTORE-commando kunnen zij opnieuw gelezen worden.

Het <masker> is analoog aan het <masker> in DISPLAY FILES. Vraagtekens maskeren bepaalde tekens en sterretjes de daarop volgende tekens.

Voorbeeld: SAVE TO <bestand> ALL LIKE MEM??X bewaart de geheugenvariabelen MEMABX, MEMFFX en MEMRRX; MEMAB, MEMABY of MAMABX echter niet.

Voorbeelden:

```
.DISPLAY MEMORY
EEN      (N)  1.0000
ALFABET  (C)  ABCDEFGHIJKL
TEKENS   (C)  ABCDEFGHIJKLMNOPQR ETC
**TOTAL**      03  VARIABLES USED      00042 BYTES USED
```

.SAVE TO MEMFILE

.RELEASE ALL

```
.DISPLAY MEMORY
** TOTAL **      00  VARIABLES USED      00000 BYTES USED
```

.RESTORE FROM MEMFILE

```
.DISPLAY MEMORY
EEN      (N)  1.0000
ALFABET  (C)  ABCDEFGHIJKL
TEKENS   (C)  ABCDEFGHIJKLMNOPQR ETC
** TOTAL **      03  VARIABLES USED      00042 BYTES USED
```

SELECT

```
SELECT [PRIMARY]
      [SECONDARY]
```

Met dit commando kiest dBASE één van de twee database-gebieden voor verdere bewerkingen. De gebruiker kan dus twee databases tegelijk in gebruik hebben; bijvoorbeeld om gegevens uit één database te gebruiken om een andere bij te werken; om twee databases met elkaar te vergelijken; of om allerlei andere bewerkingen te doen waarbij meer dan één database nodig is.

Wanneer dBASE wordt opgestart is het PRIMARY gebied actief. Dit blijft zo tot een SELECT SECONDARY-instructie gegeven wordt. Het secundaire gebied wordt dan actief totdat het systeem een SELECT PRIMARY-commando aantreft. In beide gebieden kan een andere database gebruikt worden. Op deze manier kunnen twee gegevensbanken bijna gelijktijdig gebruikt worden. Het SELECTeren van een gebied dat al geSELECTeerd was, heeft geen effect.

Wanneer beide gebieden databases in USE hebben, dan kunnen uit beide gebieden veldvariabelen gehaald worden. Een uitdrukking kan dus variabelen uit beide databasegebieden gebruiken. Als bepaalde veldnamen in beide gebieden identiek zijn, kan een "P" of "S" voor de variabele geplaatst worden om te bepalen om welke database het gaat.

dBASE-commando's die de record-pointer in de database verplaatsen (zoals GOTO, SKIP, REPORT, SORT, COPY, LIST, DISPLAY) werken alleen in de gekozen database. Met het SET LINKAGE ON-commando wordt het mogelijk om bij alle volgende commando's (die een <bereik >-parameter hebben) de recordpointers in de primaire en secundaire databases tegelijk te verplaatsen (zie SET-commando). Het REPLACE-commando werkt alleen met de variabelen in de gekozen database. Het DISPLAY STRUCTURE-commando geeft alleen de structuur van de gekozen database.

Voorbeelden:

```
.USE INKOPEN

.LIST
00001 Bonen           5           0.75
00002 Broden         2           1.06
00003 Biefstuk       4           4.33
00004 Papieren borden 1           0.94
00005 Plastic vorken  5           0.42
00006 Groente        2           0.53
00007 Kaas           1           1.96
00008 Melk           2           1.30
00009 Houtskool       2           0.75
```

130 ... SELECT

Open nu een andere database in het secundaire gebied

```
.SELECT SECONDARY
```

```
.USE INKKOST
```

```
.LIST
```

00001	840104	31.38
00002	840111	45.69
00003	840118	51.18
00004	840124	48.19
00005	840201	55.82
00006	840209	12.04
00007	840229	12.04

```
.SELECT PRIMARY
```

```
.SUM PRIJS
```

```
12.04
```

```
.SELECT SECONDARY
```

```
.APPEND
```

```
RECORD 00008
```

```
DATUM :840303
```

```
BEDRAG :12.04
```

```
RECORD 0009
```

```
DATUM : <return>
```

```
.SUM BEDRAG
```

```
268.38
```

Variabelen van beide databases beschikbaar

```
.DISP OFF PRIJS,BEDRAG,ITEM,DATUM
```

```
0.75      12.04      Houtskool      840303
```

Dezelfde database kan in beide gebieden gebruikt worden

```
.USE INKOPEN
```

Wees voorzichtig: namen van variabelen zijn dezelfde in beide databases

SET

- a. SET <parm1> [ON]
[OFF]
- b. SET <parm2> TO <opt>

Dit commando wijzigt de configuratie van dBASE. SET heeft twee vormen. Met vorm a. kunnen de ON/OFF-parameters aan- of uitgezet worden; met vorm b. krijgen de parameters die een reeks vereisen (hieronder beschreven) de gewenste waarde.

Vorm a.: parameters en standaardwaarden:

Opmerking: De standaard-systeemwaarden zijn onderstreept.

<parm1>	WERKING	BETEKENIS
1. ECHO	ON	Alle commando's uit een commandobestand worden op het scherm gebracht.
	<u>OFF</u>	Geen weergave.
2. STEP	ON	dBASE wacht na de uitvoering van een commando tot de gebruiker besloten heeft om ofwel over te gaan naar het volgende commando of het commandobestand te verlaten of een commando in te voeren vanaf het toetsenbord (STEP wordt gebruikt om fouten in commandobestanden op te sporen).
	<u>OFF</u>	Normale werking wordt hervat.
3. TALK	<u>ON</u>	De resultaten van commando's worden op het scherm weergegeven.
	OFF	Geen weergave.
4. PRINT	ON	Uitvoer wordt op de printer weergegeven.
	<u>OFF</u>	Geen weergave op de printer.
5. CONSOLE	<u>ON</u>	De uitvoer wordt op het scherm weergegeven.
	OFF	Geen uitvoer naar het scherm.
6. ALTERNATE	ON	De uitvoer gaat naar een schijfbestand.
	<u>OFF</u>	De uitvoer naar het bestand wordt uitgezet.

132 ... SET

7. SCREEN	<u>ON</u>	Full-screen werking wordt ingesteld voor APPEND, INSERT, EDIT en CREATE.
	OFF	Geen full-screen werking.
8. LINKAGE	<u>ON</u>	Alle volgende commando's (LIST, REPORT, SUM, d.w.z. alle commando's met een < bereik >), verhogen gelijktijdig de record-pointers van het primaire en secundaire gebied. LINKAGE stelt deze pointers niet gelijk als ze verschillen.
	<u>OFF</u>	De pointers voor de primaire en secundaire database worden onafhankelijk van elkaar.
9. COLON	<u>ON</u>	Begrenst de GET-gegevensitems in "@-commando's met dubbele punten.
	OFF	Geen dubbele punten.
10. BELL	<u>ON</u>	Beltoon wordt gegeven wanneer ongeldige gegevens ingevoerd worden of de grenzen van een gegevensveld overschreden worden.
	OFF	Geen beltoon.
11. ESCAPE	<u>ON</u>	Een < escape > (1B Hex) breekt de uitvoering van commandobestanden af.
	OFF	< escape > breekt de uitvoering niet af.
12. EXACT	<u>ON</u>	Vereist dat tekenreeksen precies gelijk zijn (behalve spaties achteraan) in uitdrukkingen en het FIND-commando.
	<u>OFF</u>	Reeksen worden slechts vergeleken over de lengte van de tweede reeks b.v. "ABCDEF" = "ABC" is waar.
13. INTENSITY	<u>ON</u>	Full-screen-bewerkingen gebruiken tweevoudige intensiteit (normale en inverse-video bij bepaalde terminals).
	OFF	Tweevoudige intensiteit wordt niet gebruikt.

14. DEBUG	ON	De uitvoer van ECHO- en STEP-commando's wordt naar de printer gestuurd zodat full-screen-commando's gecontroleerd kunnen worden zonder dat het scherm helemaal vol komt te staan.
	<u>OFF</u>	Geen extra uitvoer naar de printer.
15. CARRY	ON	Gegevens uit het voorgaande record worden overgenomen wanneer records met het APPEND-commando in de full-screen modus worden toegevoegd.
	<u>OFF</u>	Geen overname van gegevens.
16. CONFIRM	ON	dBASE springt pas over naar het volgende veld bij full-screen editen als een besturings-toets (zoals <return>) wordt aangeslagen.
	<u>OFF</u>	dBASE springt over naar het volgende veld zodra teveel tekens worden ingevoerd.
17. EJECT	<u>ON</u>	REPORT geeft een paginasprong bij het begin van een nieuw rapport.
	OFF	Geen paginasprong bij een nieuw rapport.
18. RAW	ON	Spaties worden weggelaten.
	<u>OFF</u>	Plaatst spaties tussen de velden als DISPLAY- en LIST-commando's zonder de veldlijst gebruikt worden.
19. DELETED	ON	Records met wismarkering worden niet gevonden door FIND-commando's en ook niet verwerkt met commando's waarin een NEXT-bepaling kan staan (b.v. LIST, LOCATE, COUNT, enz.)
	<u>OFF</u>	Records met wismarkering kunnen worden gelokaliseerd met LOCATE en weergegeven met DISPLAY (maar niet gekopieerd met COPY of toegevoegd met APPEND).

Vorm b.: parameters en hun formats:

1. SET HEADING TO <reeks>

Deze vorm van het SET-commando bewaart de <reeks> in het intern geheugen en print de reeks als een deel van de rapporttitel. De <reeks> kan hoogstens 60 tekens lang zijn. (Zie REPORT voor een voorbeeld.)

2. SET FORMAT TO [SCREEN]
[PRINTER]
[<formatbestand>]

De eerste twee vormen van deze SET-parameter bepalen waar de output van de "@"-commando's naar toe gaat. De laatste vorm bepaalt van waaruit de "@"-commando's met READ gelezen worden. (Zie de "@" en READ-commando's.)

3. SET DEFAULT TO <drive>

Met dit SET-commando wordt de gespecificeerde disk-drive de standaard-eenheid. dBASE gaat ervan uit dat bestandsnamen waarvoor de drive niet gespecificeerd werd, op deze disk-drive staan. Het voordeel hiervan is dat commandobestanden niet voor elk bestand een drive-specificatie hoeven te geven. De &-macro's dragen verder bij tot een eenvoudige drive-keuze. In de interactieve dBASE-modus laat dit SET-commando impliciete bestandsnamen toe.

Als een standaard-eenheid ingesteld is, wordt deze drive impliciet gekozen voor ALLE bestanden waarvoor geen drive is opgegeven. Dit kunnen formulierbestanden, commandobestanden, geheugenbestanden, formatbestanden, indexbestanden, tekstbestanden en database-bestanden zijn.

In de parameter <drive> mag een dubbele punt (:) staan, dus zowel "B" als "B:" zijn geldige drive-bepalingen.

Opmerking: Dit SET-commando beïnvloedt de standaard CP/M-eenheid niet. De standaard dBASE-eenheid is bij het opstarten altijd de standaard CP/M-eenheid. Met SET DEFAULT wordt alleen onder dBASE de interne standaard drive gedefinieerd.

Voorbeeld:

.SET DEFAULT TO B:

.USE DATEVSYR

(dBASE kiest eenheid 'B' voor deze database en voor alle volgende bestanden zonder drive-bepaling)

4. SET ALTERNATE TO [<bestand>]

Deze vorm van het SET ALTERNATE commando vormt een deel van een tweedelig proces, waarmee alles wat gewoonlijk alleen op het scherm geschreven wordt ook in een schijfbestand komt. De output die door dBASE gegenereerd wordt en ook wat via het toetsenbord wordt ingevoerd bijvoorbeeld, komt in dit schijfbestand. Deze vorm identificeert en opent het schijfbestand dat de gegevens ontvangt. Als het <bestand> reeds vóór dit commando op de schijf stond, wordt het overschreven. Een SET ALTERNATE ON moet volgen om het echo-proces te starten.

Voorbeeld:

```
SET ALTERNATE TO B:PRINTBST
SET ALTERNATE ON
.
.
.
commando's
.
.
.
SET ALTERNATE TO anderbst
```

Alles wat op het scherm of op de printer verschijnt, wordt (in dit voorbeeld) gekopieerd in B:PRINTBST.TXT dat daarna als tekst verwerkt, geprint of bewaard kan worden.

5. SET DATE TO dd/mm/jj

Het format mag ook jj/mm/dd, mm/dd/jj, enz. zijn. m, d en j zijn willekeurige cijfers. Met dit commando kan de systeemdatum op elk moment ingesteld of gewijzigd worden. De geldigheid van de datum wordt echter niet gecontroleerd, wat bij het opstarten van dBASE wél gebeurt.

```
SET DATE TO 10,12,82
```

6. SET INDEX TO <indexbestand>
[, <indexbestand> ,... <indexbestand>]

Met SET INDEX TO worden max. zeven indexbestanden geïdentificeerd en opgesteld die voor latere bewerkingen gebruikt kunnen worden. Als er al een indexbestand in USE is wanneer dit commando gegeven wordt, dan wordt dat oude indexbestand gesloten en wordt een nieuw opgesteld.

Opmerking: Als de nieuwe index wordt opgesteld blijft de database op dezelfde plaats, maar de indexpointer wijst naar geen enkel record. Om de indexpointer te positioneren moeten FIND- of GOTO-commando's gebruikt worden: pas daarna zullen commando's met een NEXT-bepaling werken.

Het eerst genoemde indexbestand wordt beschouwd als de Hoofdindex. De FIND-commando's gebruiken alleen dit bestand en de databasegegevens staan in de volgorde van de Hoofdindex (bij het SKIPpen).

Een SET INDEX TO-commando (zonder indexbestanden) maakt alle indexen vrij en de database werkt opnieuw als een sequentieel bestand.

7. SET MARGIN TO n

Met deze vorm van het SET-commando kan de gebruiker de linker marge bepalen bij het printen van een rapport. De regels die geprint moeten worden beginnen na n spaties. De parameter n moet een constante zijn tussen 1 en 254.

SKIP

SKIP [+ <uitdr>]
 [- <uitdr>]

Met dit commando wordt de recordpointer voor- of achteruit gezet t.o.v. zijn huidige positie.

Voorbeeld:

```
.USE VOORRAAD

.LIST
00001 136928 13 1673 SLEUTEL          7.13 189  9  0  9.98
00002 221679  9 1673 HANDZAAG         5.17 173  4  1  7.98
00003 234561  0  96  PLASTIC BUIS     2.18  27 112 53  4.75
00004 556178  2  873 KATROL           22.19 117  3  0 28.50
00005 723756 73  27  ELEK. KAST       19.56 354  6  1 29.66
00006 745336 13  27  ZEKERING        12.65  63  7  2 15.95
00007 812763  2 1673 GLOBE            5.88 112  5  2  7.49
00008 876512  2  873 IJZERDRAAD       3.18  45  7  3  4.25
00009 915332  2 1673 VIJL             1.32  97  7  3  1.98
00010 973328  0  27  DEKSEL KAN       0.73  21 17  5  0.99

.5

.SKIP -2
RECORD: 00003

.SKIP
RECORD: 00004

.SKIP 3
RECORD: 00007
```

SORT

SORT ON <veld> TO <bestand> [ASCENDING]
[DESCENDING]

Met dit commando kan de gebruiker databases sorteren in een bestand dat afwijkt van het oorspronkelijke. Het bestand in USE wordt gesorteerd op één gegevensveld en dit gebeurt in stijgende (ASCENDING) of dalende (DESCENDING) volgorde. Het USE-bestand blijft in USE, het wordt niet gewijzigd en gewisse records worden genegeerd.

Hoewel bij het SORT-commando slechts één sleutel gebruikt kan worden, kan een database door opeenvolgende SORT's toch op meerdere sleutels gesorteerd worden: men begint met de minst belangrijke sleutel en stijgt daarna stap voor stap in de hiërarchie. dBASE zal de recordvolgorde alleen veranderen als dit nodig is. De sorteervolgorde voor tekenvelden is de ASCII-code. Als noch ASCENDING, noch DESCENDING gespecificeerd is, wordt ASCENDING aangenomen.

Het sorteren gebeurt naar de ASCII-waarde van de tekens. Dit betekent dat de reeks 'SMITS' "kleiner" is dan 'Smits' (een uitdrukking " 'SMITS' < 'Smits' " zou waar zijn).

Het indexcommando is te vergelijken met het SORT-commando: INDEX voert nagenoeg alle SORT-taken uit, maar INDEX geeft meer mogelijkheden en werkt sneller dan SORT.

Met SORT wordt de database opnieuw op de schijf geschreven, u moet dus minstens zoveel schijfcapaciteit hebben als het oorspronkelijk bestand innam.

.USE INKOPEN

.LIST

00001	BONEN # 303 BLIK	5	0.75
00002	BROOD	2	0.97
00003	BIEFSTUK	4	3.94
00004	PAPIEREN BORDEN	1	0.86
00005	PLASTIC VORKEN	5	0.42
00006	GROENTE	2	0.53
00007	KAAS	1	1.96
00008	MELK (1/2 LIT.)	2	1.30
00009	HOUTSKOOL, 5# ZAKKEN	2	0.75

.SORT ON ITEM TO INKSORT
SORT COMPLETE

.USE INKSORT

.LIST

00001	BIEFSTUK	4	3.94
00002	BONEN # 303 BLIK	5	0.75
00003	BROOD	2	0.97
00004	GROENTE	2	0.53
00005	HOUTSKOOL, 5# ZAKKEN	2	0.75
00006	KAAS	1	1.96
00007	MELK (1/2 LITER)	2	1.30
00008	PAPIEREN BORDEN	1	0.86
00009	PLASTIC VORKEN	5	0.42

STORE

STORE <uitdr> TO <geheugenvar>

Met dit commando wordt de waarde van een uitdrukking berekend en in een geheugenvariabele opgeslagen. Als de geheugenvariabele nog niet bestond voordat dit commando gegeven werd, creëert dBASE hem automatisch.

STORE wijzigt en creëert alleen geheugenvariabelen. Om veldvariabelen uit een database te wijzigen, moet het REPLACE-commando gebruikt worden.

.RELEASE ALL

.STORE 1 TO EEN

1

.STORE 'ABCDEFGHijkl' TO ALFABET
 ABCDEFGHIJKL

.STORE ALFABET + 'MNOPQ ETC' TO TEKEN
 ABCDEFGHIJKLMOPQ ETC

.STORE EEN*1.0000 TO EEN
 1.0000

.DISPLAY MEMORY

EOF	(L)	.T.	
EEN	(N)	1.0000	
ALFABET	(C)	ABCDEFGHijkl	
TEKEN	(C)	ABCDEFGHIJKLMNOPQ ETC	
** TOTAL **		04 VARIABLES USED	00042 BYTES USED

SUM

SUM <veld> [<veld>][TO <geheugenvariëlijst>][<bereik>]
 [FOR <uitdr>][WHILE <uitdr>]

Met het SUM-commando worden numerieke uitdrukkingen uit het bestand in USE opgeteld volgens de <bereik>- en FOR-bepalingen. Er kunnen tot 5 uitdrukkingen tegelijk worden opgeteld. Als de TO-bepaling aanwezig is, worden de sommen ook opgeslagen in geheugenvariëbelen (als de geheugenvariëbelen nog niet bestonden voordat het SUM-commando gegeven werd, worden ze automatisch gecreëerd.) De standaard systeemwaarde voor bereik in SUM is alle records zonder wismarkering.

.USE INKOPEN

.LIST

00001	BONEN # 303 BLIK	5	0.75
00002	BRODEN	2	0.97
00003	BIEFSTUK	4	3.94
00004	PAPIEREN BORDEN	1	0.86
00005	PLASTIC VORKEN	5	0.42
00006	GROENTE	2	0.53
00007	KAAS	1	1.96
00008	MELK (1/2 LIT.)	2	1.30
00009	HOUTSKOOL, 5# ZAKKEN	2	0.75

.SUM PRIJS

11.48

.SUM PRIJS FOR AANTAL = 1

2.82

.SUM PRIJS, AANTAL

11.48 24

.SUM PRIJS TO MSOM

11.48

.?MSOM

11.48

.DISPLAY MEMORY

MSOM (N) 11.48

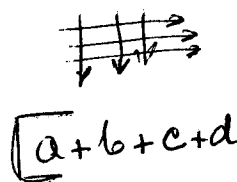
** TOTAL ** 01 VARIABLES USED 00006 BYTES USED

.? MSOM*1.10

12.6280

.SOM AANTAL*PRIJS,AANTAL,PRIJS,PRIJS/AANTAL

31.53 24 11.48 5.81



TEXT

TEXT

Alle regels die na het TEXT-commando worden gegeven, worden rechtstreeks op de printer of het scherm gekopieerd (dit hangt af van de SET-instelling). Macro's worden niet uitgebreid, maar doorlopende regels worden gecombineerd. Het systeem blijft kopiëren totdat het een ENDTEXT-commando aantreft.

Het TEXT-commando is een gemakkelijke manier om stukken tekst in commandobestanden te plaatsen, zonder daarvoor een reeks @ SAY of ?-commando's te hoeven gebruiken.

Voorbeeld:

TEXT

Alles wat tussen de TEXT en ENDTEXT verschijnt wordt naar de output gekopieerd zonder verwerkt te worden door de dBASE-interpretor. De enige uitzondering is een invoerregel die doorloopt met een puntkomma.

ENDTEXT

TOTAL

```
TOTAL ON <sleutel> TO <database> [FIELDS <lijst>]
[FOR <uitdr>]
[WHILE <uitdr>]
```

Het TOTAL-commando is ongeveer gelijk aan de subtotaliseer-functie in het REPORT-commando, alleen worden de subtotalen hier in de database geplaatst in plaats van geprint. Dit commando maakt het mogelijk relevante informatie zeer compact samen te vatten.

Opmerking: De USE-database moet van tevoren op de sleutel gesorteerd of geïndexeerd zijn.

Als de TO-database gedefinieerd is (als deze bestond met een structuur), dan blijft de structuur hetzelfde en wordt deze gebruikt om te bepalen welke velden opgeteld moeten worden.

Als de TO-database niet bestond voor dit TOTAL-commando gegeven werd, dan wordt de structuur van de USE-database naar het TO-bestand gekopieerd.

Dit commando is zeer selectief als de TO-database bestaat en de FIELD-bepaling in het commando is opgenomen. In dat geval worden alleen de numerieke velden in FIELDS opgeteld. Bij elke andere vorm van dit commando worden alle numerieke velden getotaliseerd.

TOTAL kan ook gebruikt worden om dubbele records uit een database te verwijderen aangezien niet-numerieke velden in de FIELDS-lijst (natuurlijk) niet opgeteld worden, maar ook niet als fout worden aangetekend.

Voorbeelden:

```
.USE ORDERS INDEX ORDERS
```

```
.DISP STRU
STRUCTURE FOR FILE:      ORDERS.DBF
NUMBER OF RECORDS:      00008
DATE OF LAST UPDATE:    14/12/83
PRIMARY USE DATABASE
```

FLD	NAME	TYPE	WIDTH	DEC
001	KLANT	C	020	
002	ART:NR	C	005	
003	AANTAL	N	005	
TOTAL			00031	

144 ... TOTAL

```
.LIST
00003 HARRIS, ARNOLD      11528      44
00007 JUAN, DON          21828       5
00001 SWARTZ, JOE        31415      13
00005 MACK, JAY          31415       3
00008 SALT, CLARA        70296       9
00002 SWARTZ, JOE        76767      13
00006 TERRY, HANS        76767       5
00004 ADAMS, JEAN        89793      12
```

(Stel dat men wil weten hoeveel stuks geleverd moeten worden. Door de aantallen op te tellen zolang het artikelnummer gelijk blijft, wordt een database opgesteld waarin de artikelnummers en de benodigde hoeveelheden staan.)

(De database TOTART=totaal per artikel is reeds gedefinieerd.)

```
.TOTAL ON ART:NR TO TOTART
00006 RECORDS COPIED
```

```
.USE TOTART
```

```
.DISP STRU
STRUCTURE FOR FILE:      TOTART.DBF
NUMBER OF RECORDS:      00006
DATE OF LAST UPDATE:    21/12/83
PRIMARY USE DATABASE
FLD      NAME           TYPE      WIDTH    DEC
001      ART:NR         C         005
002      AANTAL         N         005
**TOTAL**                00011
```

```
.LIST
00001      11528      44
00002      21828       5
00003      31415      16      (twee orders opgeteld)
00004      70296       9
00005      76767      18      (twee andere orders opgeteld)
00006      89793      12
```

UPDATE

```

UPDATE FROM <database> ON <sleutel> [ADD <veldlijst>]
[RANDOM]
[REPLACE [      <veldlijst>      ]]
[<veld> WITH <from-veld>]

```

Het UPDATE-commando reviseert het bestand in USE aan de hand van gegevens uit een ander bestand. Bijgewerkte gegevens kunnen ofwel opgeteld ofwel volledig vervangen worden. Een record wordt bijgewerkt wanneer een bepaald sleutelveld in de USE-database overeenkomt met een sleutelveld uit de FROM-database. Deze velden worden in de ON-bepaling gegeven.

Bij gebruik van de RANDOM-bepaling wordt ervan uitgegaan dat de <sleutel> een *enkelvoudig* veld in het FROM-bestand is dat overeen kan komen met de indexen in de USE-database (de USE-database moet geïndexeerd zijn). De records in het FROM-bestand mogen in elke volgorde staan. Bij het lezen van een record uit het FROM-bestand, zal een interne FIND het record in de USE-database opzoeken.

Als de RANDOM-optie niet gebruikt wordt, dan moet de USE-database van tevoren op de sleutel gesorteerd of geïndexeerd zijn. De FROM-database moet van tevoren op de sleutel gesorteerd zijn. Beide databases worden "teruggespoeld" en een record wordt gelezen. Als de sleutels overeenkomen, dan wordt overeenkomstig de bepaling toegevoegd of vervangen. Als de sleutel in het USE-bestand kleiner is (in sorteervolgorde) dan de sleutel uit de FROM-database, dan gebeurt er niets; het record wordt overgeslagen en blijft onveranderd. Hetzelfde gebeurt als de FROM-sleutel kleiner is: het record wordt overgeslagen en niet gewijzigd.

Voorbeeld:

```

.USE INVUPDATE
.DISPLAY STRUCTURE
STRUCTURE FOR FILE:      INVUPDAT.DBF
NUMBER OF RECORDS:      00003
DATE OF LAST UPDATE:    18/12/83
PRIMARY USE DATABASE
FLD      NAME           TYPE      WIDTH      DEC
001      ART:NR         C         005
002      VOORRADIG      N         005
003      PRIJS          N         010        002
**TOTAL**                00021

.LIST
00001                21828        77            35.88
00002                70296         0            250.00
00003                89793         2            134999.00

```

(Opmerking: de database wordt gesorteerd op de "sleutel" ART:NR)

.USE VOORRAAD INDEX VOORRAAD

```
.DISPLAY STRUCTURE
STRUCTURE FOR FILE:      VOORRAAD.DBF
NUMBER OF RECORDS:      00008
DATE OF LAST UPDATE:    18/12/83
PRIMARY USE DATABASE
FLD          NAME          TYPE      WIDTH     DEC
001          ITEM          C         020
002          PRIJS         N         010      002
003          ART:NR        C         005
004          VOORRADIG     N         005
**TOTAL**                00041
```

```
.DISP ALL
00008 KOLEN                22.00    11528    16
00005 KEUKENSTOEL        34.72    21828    77
00001 SCHAKELKLOK        9.99     24776    1
00002 BAMIBAL            1.67     31415    18
00007 RINGEN, GOUD        200.00   70296    5
00006 TROMBONES           198.37   76767    76
00004 TANK, SHERMAN       134999.00 89793    5
00003 GADGET, GROOT       16.33    92653    7
```

(Ook hier wordt de database geïndexeerd op de "sleutel" ART:NR)

```
.UPDATE ON ART:NR FROM INVUPDAT ADD VOORRADIG REPLACE
PRIJS
```

```
.LIST
00008 KOLEN                22.00    11528    16
00005 KEUKENSTOEL        35.88    21828    154
00001 SCHAKELKLOK        9.99     24776    1
00002 BAMIBAL            1.67     31415    18
00007 RINGEN, GOUD        250.00   70296    5
00006 TROMBONES           198.37   76767    76
00004 TANK, SHERMAN       134999.00 89793    7
00003 GADGET, GROOT       16.33    92653    7
```

(De twee nieuwe Sherman-tanks zijn aan de database toegevoegd en de prijs van de gouden ringen en de keukenstoel zijn door nieuwe prijzen vervangen.)

USE

```
USE [<databasebestand>]
USE <databasebestand> INDEX <indexbst>
[,<indexbst> ,...<indexbst>]
```

Voorbeeld:

```
.USE BESTAND INDEX NAAM,STAD,ART:NR,VERKOPER
```

Met het USE-commando wordt gespecificeerd welk (reeds bestaand) databasebestand het bestand in USE wordt. Als een ander bestand reeds een USE-bestand was, dan wordt het oude bestand gesloten. Ook als in het commando geen bestandsnaam gespecificeerd wordt, wordt het vorige USE-bestand gesloten.

Met de tweede vorm van USE wordt een database voor bewerking en een daarbij behorend indexbestand (dat van tevoren met het INDEX-commando gecreëerd werd) gespecificeerd. Met deze vorm zijn latere indexbewerkingen zoals FIND mogelijk en heeft men index-sequentiële toegang tot het bestand.

Er kunnen maximaal 7 indexbestanden met één database tegelijk in USE zijn. Het eerste genoemde indexbestand wordt als de Hoofdindex beschouwd. Alle FIND-commando's gebruiken alleen deze index en de databasegegevens staan in volgorde van deze Hoofdindex (bij het SKIPpen). Alle genoemde indexbestanden worden automatisch bijgewerkt wanneer de sleutels gewijzigd worden (met de commando's APPEND, EDIT, REPLACE, READ of BROWSE.)

Voorbeelden:

```
.USE VOORBEELD
```

```
.USE ORDERS INDEX ORDERS
```

148 ... WAIT

WAIT

WAIT [TO <geheugenvar>]

Met dit commando stopt dBASE met de bewerkingen en verschijnt WAITING op het scherm tot een teken wordt ingevoerd via het toetsenbord. Als de TO-bepaling gegeven wordt, wordt de toetsaanslag waardoor dBASE gaat werken, in de geheugenvariabele opgeslagen.

De TO-optie is erg handig wanneer slechts één teken nodig is om de werking van een commandobestand te regelen, b.v. bij menu-keuzes. Er is geen <return> nodig om het teken in te voeren, zoals bij de commando's ACCEPT en INPUT.

Als een niet te printen teken (bijv. <return>, <nieuwe regel> of een ander speciaal teken) ingetypt wordt als antwoord op een WAIT TO-commando, dan wordt de inhoud van de geheugenvariabele een spatie.

Voorbeeld:

```
.RELEASE ALL
```

```
.WAIT TO KEUZE  
WAITING 1
```

```
.DISP MEMO  
KEUZE (C) 1  
**TOTAL** 01 VARIABLES USED 00002 BYTES USED
```

Bilagene

BIJLAGE A COMMANDO'S

? < uitdr > [, < uitdr >]
 @ < coördinaten > [SAY < uitdr > [USING ' < picture > ']]
 [GET < variabele > [PICTURE ' < picture > ']]
 ACCEPT [" < tekenreeks > "] TO < geheugenvar >
 APPEND [FROM < bestand > [SDF] [DELIMITED] [FOR < uitdr >]]
 APPEND [BLANK]
 BROWSE [FIELDS < lijst velden >]
 CANCEL
 CHANGE [< bereik >] FIELD < lijst > [FOR < uitdr >]
 CLEAR [GETS]
 CONTINUE
 COPY TO < bestand > [< bereik >] [FIELD < lijst >] [FOR < uitdr >]
 [SDF] [WHILE < uitdr >]
 [DELIMITED [WITH < scheidingsteken >]]
 COPY TO < bestand > STRUCTURE [EXTENDED]
 [FIELD < lijst velden >]
 COUNT [< bereik >] [FOR < uitdr >] [TO < geheugenvar >]
 CREATE [< bestandsnaam >] [FROM < bestandsnaam >]
 DELETE [< bereik >] [FOR < uitdr >]
 [WHILE < uitdr >]
 DELETE FILE < bestand >
 DISPLAY [< bereik >] [FOR < uitdr >] [< lijst uitdr. >] [OFF]
 [FIELDS < lijst velden >]
 DISPLAY STRUCTURE
 DISPLAY MEMORY
 DISPLAY FILES [ON < disk-drive >] [LIKE < masker >]
 DISPLAY STATUS
 DO < bestand >
 DO WHILE < uitdr >
 EDIT [< n >]
 EJECT
 ELSE
 ENDDO
 ENDIF
 ENDTEXT
 ERASE
 FIND < tekenreeks >
 GO of GOTO [RECORD], of [TOP], of [BOTTOM], < n >
 HELP [commando]
 IF < uitdr >
 INDEX
 INDEX ON < tekenreeksuitdrukking > TO < indexbestandsnaam >
 INPUT [" < tekenreeks > "] TO < geheugenvar >
 INSERT [BEFORE] [BLANK]
 JOIN TO < bestand > FOR < uitdrukking > [FIELDS < lijst velden >]
 LIST

150 ... BIJLAGE

LOCATE [< bereik >] [FOR < uitdr >]
LOOP
MODIFY STRUCTURE
MODIFY COMMAND < commandobestand >
NOTE or *
PACK
QUIT [TO < lijst CP/M-commando's of .COM bestanden >]
READ [NOUPDATE]
RECALL [< bereik >] [FOR < uitdr >]
REINDEX
RELEASE [< geheugenvarlijst >] [ALL [LIKE < masker >]]
REMARK
RENAME < huidige bestandsnaam > TO < nieuwe bestandsnaam >
REPLACE [< bereik >] < veld > WITH < uitdr > [AND < veld > WITH
< uitdr >]
REPORT [< bereik >] [FORM < formulierbestand >] [TO PRINT] [FOR
< uitdr >]
RESET
RESTORE FROM < geheugenbestand > [ADDITIVE]
RETURN
SAVE TO < bestand > [ALL LIKE < masker >]
SELECT [PRIMARY of SECONDARY]
SET < parm > [ON] [OFF]
SET ALTERNATE TO < bestand >
SET DEFAULT TO < drive >
SET DATE TO < tekenreeks >
SET FORMAT TO < formatbestandsnaam >
SET HEADING TO < tekenreeks >
SET INDEX TO < lijst indexbestanden >
SET MARGIN TO < n >
SKIP < +/-> [< n >]
SORT ON < veld > TO < bestand > [ASCENDING], of [DESCENDING]
STORE < uitdr > TO < geheugenvar >
SUM < veld > [< bereik >] [TO < geheugenvarlijst >] [FOR < uitdr >]
TEXT
TOTAL TO < bestand > ON < sleutelvariabele > [FIELDS < lijst
velden >]
UPDATE FROM < bestand > ON < sleutelvariabele > [ADD < lijst
velden >] [REPLACE < lijst velden >]
USE < bestand > [INDEX < indexbestandsnaam >]
WAIT [TO < geheugenvar >]

FUNCTIES

@(< reeks1 > , < reeks2 >)	plaatsingsfunctie
*	gewist-record-functie
#	recordnummer-functie
!(< tekenreeks >)	hoofdletter-functie
\$(< tekenreeks > , < start > , < lengte >)	deelreeks-functie
CHR(< numerieke uitdrukking >)	getal-naar-tekenfunctie
DATE()	systeemdatum-functie
EOF	einde-bestand-functie
FILE(< bestand >)	bestandsfunctie
INT(< numerieke uitdrukking >)	geheel-getal-functie
LEN(< tekenreeks >)	lengte-functie
RANK(< tekenreeks >)	rang-functie
STR(< num. uitdr. > , < lengte > [, < decimalen >])	reeks-functie
VAL(< tekenreeks >)	waarde-functie
TRIM(< tekenreeks >)	verkortingsfunctie
TYPE(< uitdr >)	gegevenstype-functie

**BIJLAGE B LIMIETWAARDEN EN SYSTEEM-
BEPERKINGEN**

aantal velden per record	max 32
aantal tekens per record	max 1000
aantal records per database	max 65535
aantal tekens per tekenreeks	max 254
precisie van numerieke velden	10 cijfers
grootste getal	ongeveer $1,8 \times 10^{63}$
kleinste getal	ongeveer $1,0 \times 10^{-63}$
aantal geheugenvariabelen	max 64
aantal tekens per commandoregel	max 254
aantal uitdrukkingen in SUM-commando	max 5
aantal tekens in REPORT-titel	max 254
aantal tekens in indexsleutel	max 99
aantal GETS voor een READ	max 64
aantal bestanden gelijktijdig open	max 16

BIJLAGE C FOUTMELDINGEN

BAD DECIMAL WIDTH FIELD

Fout in aantal decimalen. Decimalen opnieuw definiëren.

BAD FILE NAME

Fout in bestandsnaam.

BAD NAME FIELD

Fout in veldnaam. Naam opnieuw definiëren.

BAD TYPE FIELD

Fout in veldtype. Moet C (teken), N (numeriek) of L (logisch) zijn.

BAD WIDTH FIELD

Fout in veldlengte. Veldlengte opnieuw definiëren.

CANNOT INSERT – THERE ARE NO RECORDS IN DATABASE FILE

Er kan geen record worden ingelast: er staan nog geen records in de database. Gebruik het commando APPEND.

CANNOT OPEN FILE

Interne fout: bestand kan niet worden geopend. Neem contact op met uw dealer.

COMMAND FILE CANNOT BE FOUND

Commandobestand niet op de schijf. Controleer de spelling.

DATA ITEM NOT FOUND

Gegevensitem niet gevonden.

DATABASE IN USE IS NOT INDEXED

De database in USE is niet geïndexeerd. FIND is alleen mogelijk in geïndexeerde databases.

DIRECTORY IS FULL

Er passen geen bestanden meer op het CP/M-schijfregister.

DISK IS FULL

Er is geen opslagruimte meer op de schijf beschikbaar.

END OF FILE FOUND UNEXPECTEDLY

Einde bestand onverwacht bereikt. De database in USE heeft niet het juiste format. Als alle records erin staan en juist zijn, moet u de database schonen met PACK en opnieuw indexeren met het commando INDEX.

156 ... BIJLAGE

"FIELD" PHRASE NOT FOUND

Geen FIELD-bepaling aangetroffen. Voer de commandoregel opnieuw in.

FILE ALREADY EXISTS

Bestand bestaat al.

FILE DOES NOT EXIST

Bestand bestaat niet.

FILE IS CURRENTLY OPEN

Bestand is momenteel open. Geef een commando USE of CLEAR om het bestand te sluiten.

FORMAT FILE CANNOT BE OPENED

Formatbestand kan niet worden geopend.

FORMAT FILE HAS NOT BEEN SET

Formatbestand is niet bepaald. Leg het gewenste formatbestand vast met SET.

ILLEGAL DATA TYPE

Ongeldig gegevenstype.

ILLEGAL GOTO VALUE

Ongeldige GOTO-waarde; waarde moet > 0 en < 65535 zijn.

ILLEGAL VARIABLE NAME

Ongeldige variabelenaam; in variabele- en veldnamen zijn alleen alfanumerieke tekens en dubbele punten toegestaan.

INDEX DOES NOT MATCH DATABASE

Er is geen overeenkomst tussen de sleutel en de database. Probeer een ander indexbestand.

INDEX FILE CANNOT BE OPENED

Indexbestand kan niet worden geopend. Controleer de spelling of indexeer de database met het commando INDEX.

JOIN ATTEMPTED TO GENERATE MORE THAN 65,534 RECORDS

JOIN zou te veel records tot gevolg hebben: maak de voorwaarde strikter.

KEYS ARE NOT THE SAME LENGTH

Sleutels hebben verschillende lengten.

MACRO IS NOT A CHARACTER STRING

¯o's moeten tekenreeksen zijn.

MORE THAN 5 FIELDS TO SUM

De SUM-limiet is 5 velden.

NESTING LIMIT VIOLATION EXCEEDED

Te diep genest.

NO EXPRESSION TO SUM

Geen < uitdr > bij SUM.

NO "FOR" PHRASE

Geen "FOR"-bepaling aangetroffen. Herschrijf het commando met de juiste syntaxis.

NO "FROM" PHRASE

Geen "FROM"-bepaling aangetroffen. Herschrijf het commando met de juiste syntaxis.

NO FIND

Meer een "constatering" dan een echte foutmelding. dBASE kon de sleutel niet vinden. Recordnummer (#) staat nu op 0 (nul).

NON-NUMERIC EXPRESSION

Niet-numerieke uitdrukking.

NONEXISTENT FILE

Bestand bestaat niet.

"ON" PHRASE NOT FOUND

"ON"-zin niet gevonden. Herschrijf het commando met de juiste syntaxis.

OUT OF MEMORY FOR MEMORY VARIABLES

Geen geheugen meer voor geheugenvariabelen: verklein omvang of aantal van de geheugenvariabelen.

RECORD LENGTH EXCEEDS MAXIMUM SIZE (OF 1000)

Recordlengte is groter dan maximaal toegestaan. Maak de lengte van enkele velden kleiner of creëer een tweede database met eenzelfde sleutel.

RECORD NOT IN INDEX

Record niet in index. Indexbestand werd niet bijgewerkt nadat een record werd toegevoegd. Indexeer opnieuw.

RECORD OUT OF RANGE

Record niet binnen bereik; recordnummer is groter dan het aantal records in de database. Het record bestaat niet.

SORTER INTERNAL ERROR, NOTIFY SCDP

Interne sorteerfout, neem contact op met uw dealer.

SOURCE AND DESTINATION DATA ARE DIFFERENT

Gegevenstype van bron en bestemming zijn niet hetzelfde. De gegevenstypes moeten beide numeriek of beide tekens zijn.

*****SYNTAX ERROR*****

Syntaxisfout.

SYNTAX ERROR IN FORMAT SPECIFICATION

Syntaxisfout in formatspecificatie.

SYNTAX ERROR, RE-ENTER

Syntaxisfout, voer opnieuw in.

"TO" PHRASE NOT FOUND

Geen "TO"-zin aangetroffen. Herschrijf het commando met de juiste syntaxis.

TOO MANY CHARACTERS

Teveel tekens. Maak de commandoregel korter.

TOO MANY FILES ARE OPEN

Er zijn teveel bestanden open; er mogen maximaal 16 bestanden tegelijk open zijn.

TOO MANY MEMORY VARIABLES

Teveel geheugenvariabelen. Er kunnen maximaal 64 geheugenvariabelen gebruikt worden.

TOO MANY RETURNS ENCOUNTERED

Tevaak < return > tegengekomen. Waarschijnlijk een fout in de structuur van een commandobestand.

"WITH" PHRASE NOT FOUND

Geen "WITH"-bepaling aangetroffen. Schrijf het commando opnieuw met de juiste syntaxis.

UNASSIGNED FILE NUMBER

Interne fout (niet toegewezen bestandnummer): neem contact op met uw dealer.

*****UNKNOWN COMMAND**

Onbekend commando. Controleer de spelling.

VARIABLE CANNOT BE FOUND

Variabele niet gevonden. De variabele moet nog gecreëerd worden of de spelling gecontroleerd.

Aanvullingen

Wat dit pakket bevat

Documentatie

- Nieuwe dBASE II installatie-instructies (alleen voor MS-DOS en CPM-86)
- dGEN, een systeem dat dBASE II applicatie-programma's genereert.
- dSORT, een snel-sorteerprogramma voor uw dBASE II bestanden.
- Een samenvatting van de wijzigingen in dBASE II Versie 2.41

Diskettes

- dBASE II System Disk die de volgende bestanden bevat:
 - dBASE.COM
 - dBASEOVR.COM
 - dBASEMSG.TXT (CPM-80 diskettes bevatten ook INSTALL.COM, Apple diskettes APINSTAL.COM)
 - dGEN.PRG (CPM-80 diskettes bevatten dGEN.COM)
 - dSORT.COM
 - INSTALL.COM (behalve CPM-80)
- dBASE II Sample Programs Disk

2... AANV

Gebruiksaanwijzing van uw Sample Program Diskette

1. Start uw systeem met het gebruik van de instructies in de handleiding van uw computersysteem.
2. Schuif uw dBASE II Sample Program diskette in drive B.
3. Schuif uw dBASE II diskette in drive A.
4. Bij de A > typt u
A > DBASE
5. U ziet de dBASE II copyright-boodschap en de promptstip
6. Bij de . typt u
. SET DEFAULT TO B
. DO SP-MAIN
7. Om een lijst van de programma's met voor elk een korte beschrijving te zien, kiest u item 4. uit het menu (help) op het scherm.

DBASE II SAMPLE PROGRAMS

0. exit
1. mailing labels
2. inventory program
3. checkbook program
4. help

select: :



dSORT

ASHTON · TATE ™

© 1985 Ashton-Tate
Reproduction by any method is strictly prohibited

dBASE II SORT Utility
(Voor 8-bits computers)

INHOUD

Beschrijving, Mogelijkheden, Prestaties	4
Overzicht van de procedures	5
dBASE SORT Prompt-Commandoformat	7
dBASE SORT Vormvrij Commandoformat	11
dBASE SORT Parameter-Commandoformat	13
Voorbeelden	13
Automatische controle diskettecapaciteit	16
Schermmeldingen	16

dBASE SORT – 8-BITS PAKKET

3

Beschrijving

dBASE SORT is een snel, flexibel en nuttig instrument ter sortering van databases die gecreëerd zijn in dBASE II met een CP/M besturingssysteem. Met dBASE SORT, kunt u

- (1) tot op 32 velden tegelijk sorteren, zonder een merkbare toename van de sorteertijd;
- (2) stijgende en dalende volgorden voor ieder sorteerveld specificeren;
- (3) specificeren of tekenvelden alfabetisch gesorteerd moeten worden, of, zoals de interne dBASE II sorteerfunctie, met hoofdletters vóór kleine letters;
- (4) onderzoeken of er genoeg capaciteit op de diskette beschikbaar is alvorens tot een sortering over te gaan.

Mogelijkheden

U kunt maximaal 65.535 records sorteren, met een maximale database-omvang van 8 megabytes. De sorteersleutel (de totale lengte van alle velden die in één keer gesorteerd kunnen worden) mag niet groter zijn dan 1000 bytes (reken één byte extra voor elk numeriek veld voor het berekenen van dit totaal).

Prestaties

dBASE SORT werd getest op een 64 K CP/M systeem met double density floppy drives. Het bestand had 2000 records van 80 bytes.

Records per minuut	Aantal en type van de sorteervelden	Lengte van de sorteersleutel
1000	1 teken	10 bytes

dsad

dBASE SORT – OVERZICHT VAN DE PROCEDURES**Start dBASE SORT**

1. Als u dBASE II gebruikt moet u het verlaten (met het commando QUIT). Gebruik één van de volgende commandoformats om dBASE SORT te starten:

PROMPT-Format – Typ de antwoorden op de vragen die op het scherm gesteld worden; wanneer u hiermee klaar bent begint het sorteren.

VORMVRIJ Format – Voer een éénregelig commando in dat alle sorteerdetails aangeeft, dus inclusief de naam van de database, de sorteervelden, de naam van het nieuwe (output)bestand, en de codes voor de gewenste sorteeropties.

PARAMETER-Format – Typ een gemodificeerd vormvrij commando in een afzonderlijk bestand, genaamd < bestandsnaam.PRM >. Om het sorteren te starten, voert u het commando SORT @ < bestandsnaam > in – zonder de .PRM uitbreiding.

Automatische controle van de geheugencapaciteit

2. Alvorens met sorteren te beginnen, controleert dBASE SORT of er genoeg capaciteit op de diskette beschikbaar is om de uitvoering ervan te kunnen voltooien. In het geval van “interne sorteringen”, waarbij de volledige database in het centrale geheugen past, is er slechts de ruimte voor het uitvoerbestand vereist. In andere gevallen vereist dBASE SORT tweemaal zoveel ruimte als de input-database; éénmaal voor een intermediaire werkruimte op de diskette, en dezelfde hoeveelheid nogmaals voor het uitvoerbestand.

Als er onvoldoende ruimte is, voert dBASE SORT de sortering niet uit en op het scherm verschijnt de foutmelding **d26 – INSUFFICIENT DISK SPACE ON X: – nnnnK AVAIL mmmmK REQ'D** (onvoldoende ruimte op schijf: nnnnK beschikbaar, mmmmK nodig). U kunt dan het werkbestand aan een andere drive toewijzen, en/of de te sorteren database overschrijven om zo het gebrek aan geheugencapaciteit te overwinnen.

6... AANV

Onderbreking

3. Wanneer u de sortering wilt annuleren nadat zij reeds is gestart, dan slaat u de < escape >-toets aan. Binnen 4 of 5 seconden verschijnt de volgende boodschap op het scherm:

DO YOU WANT TO ABORT? (Y/N)

Als u met 'Y' antwoordt, zal het sorteren stoppen. Als u de 'N' aanslaat zal het sorteren voortgaan waar onderbroken werd.

Antwoord niet met 'Y' als van de overschrijfoptie gebruik is gemaakt. Het inputbestand zou in zo'n geval onbruikbaar zijn.

nn% SORTED

4. De melding % SORTED vertelt u hoeveel van het sorteerproces voltooid is. "Interne sorteringen" geven geen % SORTED, daar hun uitvoering minder dan 60 seconden duurt.

Complete

5. SORT COMPLETE verschijnt altijd op het scherm na de geslaagde voltooiing van een sortering. De nieuwe database kan gebruikt worden als elke andere database die in dBASE II gecreëerd is.

dBASE SORT – PROMPT-COMMANDOFORMAT**Gebruik**

Om gebruik te maken van het PROMPT-COMMANDOFORMAT typt u **dSORT** en drukt u op **<return>**. Er zal een serie vragen gesteld worden. Telkens wanneer u de vraag beantwoordt en op **<return>** drukt, verschijnt de volgende vraag. Wanneer u al de betrokken vragen beantwoord heeft, start het sorteren automatisch.

Voorbeeld

De hele serie vragen wordt hieronder getoond, met voorbeelden van antwoorden in vet. De getoonde voorbeelden zouden een lijst produceren van alle cheques, door een klein bedrijf uitgeschreven, alfabetisch gerangschikt op de namen van de begunstigen. Elk record in de UIT-database vertegenwoordigt één cheque die uitgegeven is.

```
A > dSORT
*** dSORT Ver 2.41 Copyright (C) 1983 RSP Inc.
(Note: Use ^C to abort)
```

- (1) ENTER NAME OF DATABASE TO BE SORTED: **UIT**
- (2) ENTER NEW NAME FOR SORTED DATABASE (or <return>):
A:BEGUNST
- (3) ENTER FIELD NAME(S) TO CONTROL SORT (? for info):
BEG:NAAM
- (4) ANY SORT OPTIONS ? (Y/N): **Y**
(For default press <return> for each question.)
- (5) ENTER OPTIONAL WORKDRIVE: **B**
- (6) FORCE CAPITALS ? (Y/N): **Y**
- (7) OVERWRITE INPUT DATABASE ? (Y/N): **N**
- (8) RETAIN DELETED RECORDS ? (Y/N): **N**

Commentaar bij het scherm

De PROMPT-vragen in de afbeelding van het scherm worden in de onderstaande lijst in detail uitgelegd. (Opm. ^C staat voor <control>-C.)

- (1) VOER DE NAAM IN VAN DE TE SORTEREN DATABASE. Geef de drive-identificatie voor het bestand (A: of B: kunnen gebruikt worden). Laat de uitbreiding .DBF weg.

8 ... AANV

- (2) VOER NIEUWE NAAM IN VOOR DE GESORTEERDE DATABASE
(of < return >)

Standaard

Geef alleen een < return > wanneer u de naam van de input-database wenst te gebruiken (UIT in het voorbeeld). U kunt de input-bestandsnaam typen of een < return > aanslaan met hetzelfde resultaat.

WAARSCHUWING

Wanneer u een < return > aanslaat als antwoord op prompt 2, overschrijft u de input-database, zelfs als u met NO zou antwoorden op prompt 7.

Optie

Voer een nieuwe naam in voor de nieuwe database met gebruik van de drive-identificatie (A: of B:). Geen uitbreiding .DBF.

- (3) VOER SORTEERVELD(EN) IN (? voor info):

Spelling

Indien u de exacte spelling van de veldnamen voor de gewenste sorteerveld(en) niet kent, slaat u ? aan en < return >. Alle velden in de database zullen uitgelijst worden zoals het onderstaand voorbeeld toont. Daarna wordt de oorspronkelijke vraag herhaald.

(Put (D) after a field name for descending seq)

** DATABASE = A:UIT REC COUNT = 642 REC LENGTH = 80**

FIELD	TYPE	LEN	FIELD	TYPE	LEN	FIELD	TYPE	LEN
BEG:NAAM	C	30	BESCHR	C	25	CHQ:DATUM	C	8
CHQ:NUM	C	4	DOEL	C	4	BEDRAG	N	8

ENTER FIELD NAME(S) TO CONTROL SORT (? for info):

Volgorde

Typ de sorteerveldnamen in volgorde van belangrijkheid voor het sorteren; een spatie tussen elke sorteerveldnaam. Sla < return > aan.

Typ D na de veldnaam als u dat veld in dalende in plaats van in stijgende (of alfabetische) volgorde gesorteerd wenst.

(4) SORTEEROPTIES ? (Y/N):

Standaard

In het geval u geen enkele van de sorteeropties wenst. Toets N of < return > in om het sorteren te starten.

Optie

Toets Y in om de vragen t.a.v. de sorteeropties op te roepen. Nadat u Y ingevoerd heeft voor de door u gewenste opties, kunt u met de < escape > -toets het sorteren onmiddellijk laten beginnen.

(5) VOER DE OPTIONELE WERKDRIVE IN

Standaard

Geef een < return > in als dBASE SORT genoeg ruimte heeft om zijn sorteerwerkbestand (het intermediaire bestand) op dezelfde drive te creëren als die van de input-database. Op het scherm verschijnt dan de code voor de standaard-drive.

Optie

Om voor het intermediaire bestand een andere drive te gebruiken, voert u daarvan de code in en toetst u een < return >. Slechts de drives A t/m P worden geaccepteerd.

(6) OMZETTEN IN HOOFDLETTERS? (Y/N)

Standaard

Toets N of een < return > in om dezelfde sorteervolgorde te gebruiken als in dBASE II SORT standaard is. Hoofdletters worden voor de kleine letters gesorteerd. Bijvoorbeeld: ABC, ACE, ART, Abyss, Airline.

Optie

Toets Y in als u kleine letters gesorteerd wenst als zouden het hoofdletters zijn. Bijvoorbeeld: ABC, Abyss, Ace, Airline, ART.

10 ... AANV

(7) DE INPUT-DATABASE OVERSCHRIJVEN ? (Y/N):

WAARSCHUWING

Maak een reservekopie van uw input-database *voordat* u die overschrijft.

Standaard

Toets N of een < return > in om dBASE SORT de beschikking over 2 werkgebieden te geven, die elk de omvang van de oorspronkelijke database hebben. Eén ervan wordt gebruikt voor (intermediair) sorteerwerk, de andere voor de uiteindelijke output.

Optie

Toets Y in – Maak van deze optie slechts gebruik als er onvoldoende werkruimte is om de sortering uit te voeren met de 2 werkgebieden zoals hierboven beschreven. De optie voor het overschrijven van de input-database dwingt dBASE SORT tot het gebruik van slechts één werkgebied van de omvang van de oorspronkelijke database.

Om met een werkgebied minder te kunnen volstaan, moet de sorteeroperatie de input-database *overschrijven* nadat de records ervan gelezen zijn en overgebracht naar het intermediaire werkbestand. Het gesorteerde outputbestand bezet dan de ruimte die oorspronkelijk aan het inputbestand was toebedeeld.

Alvorens het sorteren te starten, vertelt dBASE SORT u dat INPUT-DATABASE ZAL NA LEZING Vernietigd worden – WILT U DOORGAAN? (Y/N). Wanneer u geen reservekopie van de inputdatabase gemaakt heeft, sla dan N aan om het sorteren geen doorgang te laten vinden en alsnog een kopie te maken. Doet u dit niet, dan zou een stroomstoring of wat voor onderbreking van het sorteerproces dan ook uw enige kopie van de database vernietigen.

(8) GEWISTE RECORDS BEHOUDEN ? (Y/N)

Standaard

Sla N of < return > aan en alle records die voor wissen gemarkeerd zijn zullen tijdens het sorteerproces verwijderd worden.

Optie

Sla Y aan en alle records die voor wissen gemarkeerd zijn zullen naar het outputbestand geschreven worden volgens de sorteersleutel, waarbij ze hun wismarkering behouden.

dBASE SORT – VORMVRIJ COMMANDOFORMAT**Gebruik**

Ter gebruik van het vormvrij commandoformat

1. Voer alle sorteerdetails in een éénregelig commando in. Hier volgen twee voorbeelden; het format wordt daaronder en op de volgende pagina uitgelegd.

A > dSORT Uit ON Chq:num; cheques voor fisc jaar 83.

A > dSORT Uit TO B:begunst ON Beg:Naam Chq:Num (D) \$WB C

2. Geef < return > .
3. Indien alle details aanvaardbaar zijn voor dBASE SORT, zal het sorteren onmiddellijk starten. Indien niet, dan verschijnt er een foutmelding.

Format

Het vormvrije format wordt hierna getoond. Iedere uitdrukking mag in hoofd- of kleine letters en moet door één of meer spaties en/of een komma gescheiden worden. Indien u commentaar wenst toe te voegen voert u een ; in aan het eind van het commando. Al hetgeen op de ; volgt wordt als commentaar gezien.

**A > dSORT < inputbestand > [TO < outputbestand >] ON
< lijst velden > [\$ < opties >]**

< inputbestand > Een nieuwe naam voor het inputbestand. Indien niet vermeld, wordt de oorspronkelijke naam behouden.

< outputbestand > De naam voor het nieuwe (gesorteerde) bestand. Het gesorteerde outputbestand wordt altijd geconstrueerd in een bestand genaamd < inputbestand > .O\$\$\$. Na de sortering krijgt dit bestand de nieuwe naam < outputbestand > .

Indien geen nieuwe naam wordt opgegeven, wordt het inputbestand gewist en het intermediaire bestand < inputbestand > .W\$\$ genoemd. Na de sortering wordt dit bestand herbenaamd met de naam van het < inputbestand > . Ook zullen de records in het outputbestand henummerd worden volgens de nieuwe volgorde van hun sortering.

< lijst velden > Lijst van sorteervelden op volgorde van belangrijkheid. Het eerste veld is van de eerste orde, het volgende van de tweede, enz. Scheid alle namen door komma's en/of spaties. Elke naam kan gevolgd worden door een (D) om een dalende volgorde voor dat veld aan te geven.

12 ... AANV

Wanneer u alle velden in de database op het scherm wenst te laten verschijnen, voert u dSORT <inputbestand> in, zonder andere uitdrukkingen; er zal geen sortering uitgevoerd worden.

<opties> De sorteropties. Als u één of meer sorteropties wenst te specificeren, voer dan een \$ in gevolgd door de gepaste code(s). Meervoudige optiecodes kunnen worden aaneengeschreven of gescheiden door spaties.

< opties >

De codes voor de sorteer-optie die in het vormvrije format worden gebruikt corresponderen met de vragen betreffende de sorteropties, gesteld in het PROMPT-format. Deze opties zijn uitvoeriger uitgelegd op de bladzijden 4 en 5.

Code Prompt-format – Vraag en commentaar

- Wx** (5) **VOER DE OPTIONELE WERKDRIVE IN** . Het bestand voor het sorteerwerk wordt normaal gecreëerd op de standaarddrive. Wilt u een andere drive gebruiken, voer dan in plaats van de x diens code (A - P) in. De drivecode moet, zonder teken of spatie, direct volgen op de W.
- C** (6) **OMZETTEN IN HOOFDLETTERS ? (Y)**. Wanneer deze optie gebruikt wordt, verandert dSORT tijdens het sorteerproces alle velden in hoofdletters; de verandering is niet permanent.
- O** (7) **INPUT VAN DE DATABASE Overschrijven ? (Y)**. Wanneer de overschrijvingsoptie wordt gebruikt, verwijst dan het TO <outputbestand> vooral niet naar een andere drive dan die van het inputbestand. Het uiteindelijke gesorteerde bestand zal het <inputbestand> vervangen en dezelfde naam hebben.
- A** (8) **GEWISTE RECORDS BEHOUDEN ? (Y)**.

dBASE SORT – PARAMETER-COMMANDOFORMAT**Format & Gebruik**

Ter gebruik van het parameter-commandoformat A > dSORT @ < parm-bestand >, gaat u als volgt te werk:

1. Gebruik een tekstverwerker voor het creëren van een “parameterbestand” dat sorteerinstructies bevat. Gebruik het vormvrij commandoformat, maar begin dit commando met de term SORT i.p.v. de term dSORT.
2. Noem het bestand < parmbestand >.PRM – bijv.: SORTEER.PRM.
3. Als het sorteren kan beginnen, roep dan het parameterbestand op door een commando in het parameter-format aan te brengen. Gebruik niet de uitbreiding .PRM. In het voorbeeld A > dSORT @B:SORTEER bevat het bestand B:SORTEER het gemodificeerde vormvrije SORT-commando.

VOORBEELDEN**Database**

In deze voorbeelden vertegenwoordigt ieder record in de UIT-database een cheque.

VOORBEELD NR. 1 * BEGUNSTIGDEN *****

Beschrijving

De nieuwe database representeert alle cheques, alfabetisch gerangschikt op naam van de begunstigde. Door het LIST commando op de nieuwe database toe te passen, creëert u dus een lijst op naam van de begunstigde en zult u snel kunnen nagaan welke cheques er uitgeschreven zijn naar bijv. J&R Supply. Omdat de FORCE CAPITALS-optie in gebruik is, zal de sortering “telefoonbestandsschikking” volgen.

Vormvrij Format

A > dSORT Uit TO BEGUNST ON BEG:NAAM \$C

```

ENTER NAME OF DATABASE TO BE SORTED:  UIT
ENTER NEW NAME FOR SORTED DATABASE (or < RETURN >):
      A:BEGUNST
ENTER FIELD NAME(S) TO CONTROL SORT (? for info):  BEG:NAAM
ANY SORT OPTIONS ? (Y/N):  Y
(For default press < RETURN > for each question.)
ENTER OPTIONAL WORK DRIVE:  A
FORCE CAPITALS ? (Y/N):  Y
OVERWRITE INPUT DATABASE (Y/N):  < ESCAPE >

```

14 ... AANV

VOORBEELD NR. 2 *** BEDRAGEN ***

Beschrijving

De nieuwe database representeert alle uitgeschreven cheques, in een numerieke volgorde waarbij het hoogste bedrag vooraan staat. Met behulp van het commando kunt u snel die begunstigde terugvinden, aan wie u bijv. een cheque van 333 gulden heeft uitgeschreven. Omdat de ruimte beperkt is, wordt de werkruimte op drive B gekozen.

Vormvrij Format

A > dSORT Uit TO Bedragen ON Bedrag (D) \$WB

```
ENTER NAME TO DATABASE TO BE SORTED:  UIT
ENTER NEW NAME FOR SORTED DATABASE (or
< RETURN >):  A:BEDRAGEN
ENTER FIELD NAME(S) TO CONTROL SORT (? for info):  BEDRAG
ANY SORT OPTIONS ? (Y/N):  Y
(For default press < RETURN > for each question)
ENTER OPTIONAL WORK DRIVE:  B
FORCE CAPITALS ? (Y/N):  < ESCAPE >
```

VOORBEELD NR. 3 *** FISCUS ***

Beschrijving

Dit creëert een nieuwe database die alle uitgaven naar soort rangschikt bijv. Huur, Comm, Div. Onder soort is BEGUNST alfabetisch op een lijst gezet. Wanneer meer dan één cheque aan dezelfde begunstigde betaald is en met dezelfde soort dan wordt de oudste (die met het laagste CHEQUENUMMER) voor de meer recente cheques gezet.

Vormvrij Format

A > dSORT Uit TO FISCUS ON Doel Beg:naam Chq:Num(D)

```
ENTER NAME OF DATABASE TO BE SORTED:  UIT
ENTER NEW NAME FOR SORTED DATABASE (or
< RETURN >):  A:FISCUS
ENTER FIELD NAME(S) TO CONTROL SORT:
DOEL BEG:NAAM CHQ:NUM(D)
ANY SORT OPTIONS ? (Y/N):  N
```

VOORBEELD OUTPUT

De structuur van de database – de volgorde van de kolommen die verschijnen wanneer het LIST-commando op de database wordt toegepast – verandert niet. Wanneer zij gerangschikt wordt naar datum, chequenummer, begunstigde, doel, en bedrag, dan heeft de nieuwe database dezelfde structuur. Alleen staan nu de records in de nieuwe volgorde.

Kolomtitels zoals hieronder getoond worden (gebaseerd op het FISCUS – Voorbeeld Nr. 3) verschijnen in werkelijkheid niet op een dBASE II lijst, maar zijn hier ter verduidelijking toegevoegd.

Record	Chq:datum	Chq: Num	Beg:Naam	Doel	Bedrag
00001	03/10/84	638	Bob Berkel	Comm	245.21
00002	26/09/84	625	Bob Berkel	Comm	260.08
00003	15/09/84	616	Bob Berkel	Comm	294.12
00004	03/10/84	634	Jan Jansen	Coomm	481.63
00005	26/09/84	626	Jan Jansen	Comm	513.27
00006	01/10/84	638	Compushop	Div	692.50
00007	02/10/84	639	Piet Kal	Huur	56.32
00008	02/10/84	640	Compushop	Kant	48.51
00009	18/09/84	620	IRGB	Mach	12.14

Resultaat in bestand FISCUS:

Primaire Rangschikking: Soort Comm, dan Div, dan Huur, dan Kant, dan Mach

Sekundaire Rangschikking: Beg:Naam Binnen het Comm-kader, wordt Bob Berkel voor Jan Jansen gecatalogiseerd

Tertiaire Rangschikking: Chq:Num(D) Voor Bob Berkel eerst 638, dan 625, dan 616.

VOORBEELD NR. 4 ***

FISCUS

Beschrijving

Dit voorbeeld toont hetzelfde als voorbeeld nr. 3, doch daarnaast illustreert het hoe u het parameter-commandoformat samen met het commando QUIT TO in dBASE II kunt gebruiken. Het bestand AUTOSORT.PRM bevat de commandoregel:

```
SORT Uit TO Fiscus ON Doel Beg:Naam Chq:Num(D) $WB
```

In dBASE II schrijft u:

```
.QUIT TO "DSORT @AUTOSORT", "DBASE"  
.USE TAXMAN
```

Het commando QUIT TO sluit alle database-bestanden af die op dat moment open zijn; dBASE SORT ondervindt dus geen problemen tijdens het sorteren van het bestand.

AUTOMATISCHE CONTROLE DISKETTE-CAPACITEIT

In de procedure "Auto Disc Space Check" zoals beschreven op pagina 3 kan met een bepaalde drive een fout optreden. Om dit te voorkomen, gebruikt u de Dynamische Debugging Tool (DDT) in CP/M om de lokatie 113B in het dSORT.COM programma van 00 naar 01 te brengen en dan het programma als volgt te bewaren:

```
A > DDT dSORT.COM      OPM: De input van de gebruiker is
      DDT VERS 2.2      onderstreept
NEXT PC
3000 0100
      -S113B
      113B 00 01
      113C 00 . <RETURN >
      -GO

A > SAVE 47 SORT.COM
```

SCHERM Meldingen

DATABASE x:filename ALREADY EXISTS – OVERWRITE ? (Y/N): De naam die u voor de nieuw gesorteerde database heeft gekozen bestaat reeds op de opgegeven drive. Voer Y in om het oude bestand te overschrijven, of N om het programma te beëindigen zodat u een nieuwe naam kunt kiezen. Er wordt van het bestandstype .DBF uitgegaan.

INPUT DATABASE WILL BE DESTROYED AFTER BEING READ. DO YOU WANT TO PROCEED ? (Y/N): Deze waarschuwing volgt altijd wanneer u de optie \$O (Overwrite) gebruikt.

SORT COMPLETE. Deze boodschap verschijnt na iedere geslaagde sortering.

DO YOU WANT TO ABORT ? (Y/N): Deze boodschap verschijnt tijdens de sortering als u de < escape >-toets aanslaat. Als u met Y antwoordt, eindigt het sorteren. Antwoordt U met N dan gaat de sortering daar verder waar zij afbrak.

nn% SORTED. Dit vertelt u hoeveel % van het sorteerproces voltooid is; het laat u weten dat het sorteren aan de gang is en geeft u een indruk van het moment van de voltooiing. Deze boodschap verschijnt niet bij "interne sorteringen", daar deze minder dan 20 seconden in beslag nemen.

AN UNDETERMINED ERROR OCCURRED –x. Er is een ernstige programmafout gebleken. Als deze opnieuw optreedt, neem dan contact op met uw leverancier.

d01 DO NOT SPECIFY FILE TYPE, .DBF OR .PRM ALWAYS ASSUMED. Dit maant u het bestandstype niet te specificeren; er wordt altijd van .DBF of .PRM uitgegaan.

- d03 **EXPECTED OPERAND MISSING.** De computer verwacht een stuurgegeven hetgeen ontbreekt.
- d04 **KEYWORD "TO" EXPECTED, NOT FOUND.** De computer verwacht het sleutelwoord "TO", doch dit blijkt niet (juist) te zijn ingevoerd.
- d05 **KEYWORD "ON" EXPECTED, NOT FOUND.** U moet op zijn minst één naam van een sorteerveld noemen.
- d06 **SORT INPUT FILE CANNOT BE FOUND ON SPECIFIED DRIVE.** U heeft wellicht de verkeerde drive opgegeven of de naam van de database fout gespeld.
- d07 **SORT INPUT FILE IS NOT A VALID dBASE .DBF FILE.** Wanneer het hier een dBASE II database betreft dan is zij beschadigd.
- d08 **NO SORT FIELDS WERE SPECIFIED.** U moet altijd op zijn minst één sorteerveld specificeren.
- d09 **INVALID FIELD NAME SPECIFIED.** Als u de namen van de sorteervelden juist gespeld heeft kan het zijn dat foutief het sleutelwoord ON voor het sleutelwoord TO werd ingevoerd. Controleer uw commando-format.
- d10 **LOGICAL FIELDS CANNOT BE SORTED.** dBASE SORT sorteert slechts numerieke velden en die van het type teken.
- d11 **INVALID SORT OPTION SPECIFIED.** Dit wordt ook gemeld als de code van de drive niet direct aansluit op de W (zonder spaties).
- d12 **INVALID DISK DRIVE CODE SPECIFIED.** Voor de codes van de drives mag men slechts de letters A t/m P gebruiken.
- d13 **MORE THAN 32 FIELDS WERE SPECIFIED.** Overschrijding van de toegestane hoeveelheid (32) veldspecificaties.
- d14 **.PRM FILE NOT ON SPECIFIED DRIVE.** Het .PRM-bestand bevindt zich niet in de gespecificeerde drive.
- d15 **.PRM FILE EMPTY OR DOES NOT CONTAIN "SORT" COMMAND.** Het SORT-commando in het parameter-bestand moet beginnen met het woord SORT, zelfs wanneer u het gebruikelijke commando veranderd heeft in een andere naam dan SORT.
- d17 **COMBINED SORT KEY LENGTH CANNOT EXCEED 1000 BYTES.** De totale lengte van de sorteersleutel mag niet meer dan 1000 bytes zijn.

18 ... AANV

- d20 **OVERWRITE OPTION INVALID WHEN FILE 2 ON DIFFERENT DRIVE.** Wanneer u de optie \$O (Overschrijving) gebruikt kunt u het beste geen naam van een outputdatabase specificeren. Als u het wel doet geldt deze slechts wanneer u dezelfde drive opgeeft als voor de inputdatabase. Deze laatste wordt in elk geval vernietigd.
- d23 **INPUT FILE IS TOO LARGE FOR THIS VERSION OF dBASE SORT.** Het inputbestand mag niet meer dan 65.535 records bevatten.
- d24 **FILE NAME CONTAINS INVALID CHARACTERS.** Bestandsnamen mogen uit niet meer dan 8 tekens bestaan en mogen geen speciale tekens bevatten zoals ., ;, : [] < > = /.
- d25 **INPUT FILE IS EMPTY.** In de opgegeven database zijn geen records aanwezig.
- d26 **INSUFFICIENT DISK SPACE ON x: - nnnnK AVAIL mmmmK REQ'D.**
x = code van de drive, nnnn = de hoeveelheid beschikbare diskettecapaciteit op deze drive, en mmmm = de hoeveelheid diskettecapaciteit die op deze drive nodig is om de sortering te voltooien.
- d28 **EXTRANEIOUS INFORMATION FOUND.** In PROMPT modus. De PROMPT-vraag wordt herhaald. U kunt slechts de informatie invoeren waarom verzocht wordt.

APPENDIX
8-BITS CP/M OP COMPUPRO

Gebruik van: BIG.DBF
225 Bytes
16 Velden

SORTERING OP 1 VELD

Veldnaam	Veldlengte	dBASE SORT	dBASE II
C:ZIP	9 tekens	43-48 sec. (267 rec/min)	3 min. 50 sec. (52 rec/min)
STATUS	14 tekens	48-50 sec. (245 rec/min)	4 min. 15 sec. (47 rec/min)
CREDITOR	30 tekens	47-48 sec. (255 rec/min)	4 min. 20 sec. (46 rec/min)

dBASE SORT gaat 5 keer sneller dan de dBASE II sorteerfunctie.

dBASE SORT – Meervoudige Velden

Lengte van de Sorteersleutel	Aantal Sorteervelden	Beschrijving van de Sorteervelden	Tijd voor dBASE SORT
32	3	4, 8, 20 tekens	48 sec. (250 rec/min)
49	6	2, 20, 10, 9, 8 tekens	67 sec. (182 rec/min)
57	6	9 cijfers, 14, 4, 2, 20, 8 tekens	65 sec. (185 rec/min)
			200 225-BYTE



dGEN

ASHTON · TATE ■TM

© 1985 Ashton-Tate
Reproduction by any method is strictly prohibited

dBASE II UTILITY dGEN

Inhoud

Samenvatting/Programma-beschrijving	1
Gereserveerde Woorden/dGEN-gebruiksaanwijzing	2
Wijzigingsprogramma's (FILEGEN)	6
Gebruiksaanwijzing van de nieuwe Wijzigingsprogramma's	8
Rapporteurprogramma (FORMGEN)	12
Adreslabelprogramma (LABELGEN)	15
Het aanmaken van Hoofdmenu's (MENUGEN)	18
Alleen voor Programmeurs	21

dGEN Samenvatting

dGEN vereenvoudigt het creëren en laten lopen van programma's voor het gebruik van een dBASE II-database. Voor het creëren van programma's hoeft u slechts de vragen te beantwoorden die dGEN op het scherm brengt.

Programma's

Men onderscheidt de volgende dGEN-programma's

- (1) dGEN – Levert het dGEN-Hoofdmenu. Hier hoeft u slechts één toets aan te slaan om binnen te treden in de hieronder beschreven utiliteitsprogramma's. U kunt deze ook direct oproepen door na de dBASE II-prompt (een punt) een DO-commando te geven.
- (2) FILEGEN – Creëert een hoofdmenu en een groep programma's ter wijziging van een bestaande database. Dankzij de opties op het nieuwe menu kunt u records in uw database bekijken, toevoegen, wijzigen of schonen (PACK).
- (3) FORMGEN – Creëert een menu en programma's voor het maken van rapporten vanuit uw database. Het menu en de rapporten zijn gelijk aan die welke u verkrijgt met het dBASE II REPORT-commando.
- (4) LABELGEN – Creëert een programma voor het maken van adreslabels van uw database. Ook nuttig voor de weergave en het afdrukken van de inhoud van een database met grote records.
- (5) MENUGEN – Creëert menu's voor programma's die binnen dBASE II geschreven zijn. Dit resulteert in wezenlijke tijdsbesparingen voor dBASE II programmeurs. Alvorens gebruik te maken van MENUGEN, moet u in staat zijn kleine wijzigingen in dBASE II programma's aan te brengen, daar de gegenereerde programma's zullen eisen dat u kleine hoeveelheden code toevoegt.

2 ... dGEN

Gereserveerde woorden

De onderstaande woorden zijn uitsluitend gereserveerd voor het gebruik door dGEN. Gebruik ze niet in uw bestands- of veldnamen.

char	filesystem	line	select
choice	formfile	linecount	stackcount
col	heading	lmargin	string
col:hdg	indexname	Mcontents	subfield
col:opts	infile	Mwidth	substack
colcount	issubtotal	option	subtotal
counter	istotal	outfile	totalopts
datafile	item	pagehdg	totstack
equals	keyfield	pagelen	width
expanded	keyvalue	pagewidth	word
expression	lastline	pos	y:n
extension	length	prompt	yourname
field:save			

dGEN GEBRUIKSAANWIJZING

Procedures

dGEN en de door haar gecreëerde programma's lopen binnen dBASE II en werken alleen met een database die gecreëerd is in dBASE II. Er zijn geen records nodig; u kunt ze later toevoegen door de wijzigingsprogramma's die u creëert met het programma FILEGEN.

.DBF-Creatie

1. Om een duplicaat te maken van de database waarvan in de volgende voorbeelden gebruik wordt gemaakt, gaat u als volgt te werk:

OPMERKING

< return > betekent: sla de return-toets aan.

A > DBASE < return >

PLEASE ENTER TODAY'S DATE DD/MM/YY OR RETURN FOR NONE
***DBASE II VERSION 2.41 1 February 1984

.CREATE < return >

ENTER FILENAME: ZOEK < return >
ENTER RECORD STRUCTURE AS FOLLOWS:

FIELD	NAME, TYPE, WIDTH, DECIMAL, PLACES
001	PERSOON,C,25 < return >
002	AFDELING,C,30 < return >
003	BEDRIJF,C,30 < return >
004	STRAAT,C,25 < return >
005	STAD,C,20 < return >
006	LAND,C,20 < return >
007	POSTCODE,C,9 < return >
008	DATUM,C,5 < return >
009	COMMENTAAR,C,25 < return >
010	< return >

INPUT DATA NOW? N < return >

Weergave structuur

2. Voor alle gebruikersprogramma's anders dan MENUGEN, moet u de veldnamen van de database exact invoeren zoals ze hier gespeld staan.

Voor een dBASE II lijst voert u de commando's in die hieronder getoond worden. Wanneer u geen gedrukte versie van de lijst wilt hebben, laat u de commando's **set print on** en **set print off** weg.

Onthoudt: < return > betekent: sla de < return > -TOETS aan.

4 ... dGEN

.DISPLAY STRUCTURE < return >
STRUCTURE FOR FILE: ZOEK.DBF
NUMBER OF RECORDS: 00000
DATE OF LAST UPDATE: 17/03/84

PRIMARY USE DATABASE

FLD	NAME	TYPE	WIDTH	DEC
01	PERSOON	C	025	
02	AFDELING	C	030	
03	BEDRIJF	C	030	
04	STRAAT	C	025	
05	STAD	C	020	
06	LAND	C	020	
07	POSTCODE	C	009	
08	DATUM	C	005	
09	COMMENTAAR	C	025	
**	TOTAL **		00190	

Toegang

3. U kunt zich tot dGEN toegang verschaffen vanuit de systeem-prompt of vanuit de dBASE II-prompt.

Vanuit het Systeem	Vanuit dBASE II	Resultaat
A > dBASE DGEN	.DO DGEN	DGEN Hoofdmenu
A > dBASE FILEGEN	.DO FILEGEN	Direct naar FILEGEN
A > dBASE FORMGEN	.DO FORMGEN	Direct naar FORMGEN
A > dBASE LABELGEN	.DO LABELGEN	Direct naar LABELGEN
A > dBASE MENUGEN	.DO MENUGEN	Direct naar MENUGEN

Hoofdmenu

4. Als U A > DBASE DGEN invoert of .DO DGEN, verschijnt het volgende op uw scherm:

DGEN MAIN MENU

- 0. EXIT
- 1. MENU GENERATOR
- 2. FILE GENERATOR
- 3. REPORT FORM GENERATOR
- 4. LABEL GENERATOR

SELECT:2:

Bij dit menu slaat u het nummer voor het door u gewenste programma aan – of 0 (nul) voor de dBASE II-prompt.

Elk dGEN programma wordt hierna uitgelegd en wel in de volgorde van het dGEN MAIN MENU. Echter, de MENU-generator – die enige programmeer-kennis vereist – komt het laatst aan de orde. Met de andere opties zal een beginnend gebruiker van dBASE II snel vertrouwd raken.

HOE MEN WIJZIGINGSPROGRAMMA'S AANMAAKT

FILEGEN

FILEGEN genereert een groep dBASE II-programma's waarmee u records in een geïndexeerd bestand kunt bekijken (VIEW), toevoegen (ADD), wijzigen (EDIT) en "schonen" (PACK). Om die nieuwe programma's te creëren, antwoordt u op prompts die FILEGEN op het scherm brengt.

Als een antwoord onaanvaardbaar is voor FILEGEN verschijnt er een foutmelding, gevolgd door een open regel en de dBASE II prompt. Voor een nieuw begin voert u in: `.DO FILEGEN <return>`.

Toegang

1. U start FILEGEN het snelst door de invoer van:

`A > DBASE FILEGEN` – na de systeemprompt of
`.DO FILEGEN` – na de dBASE II-prompt.

Wellicht verkiest u dGEN op te roepen om dan via optie 2 FILEGEN te starten. Om het Hoofdmenu te bereiken, voert u `A > DBASE DGEN` in of `.DO DGEN`.

Prompts

2. FILEGEN brengt de kop en eerste prompts op het scherm voor de "View, Add, Edit, en Pack" programma-generator.

VIEW, ADD, EDIT & PACK GENERATOR DD/MM/JJ

- (a) ENTER DATABASE FILENAME :ZOEK
- (b) ENTER INDEX FILENAME :ZOEK
- (c) ENTER INDEX KEYFIELD :BEDRIJF
- (d) INDEX FILE DOES NOT EXIST. CREATE IT? (Y/N) Y
- (e) CREATING INDEXFILE ...
GETTING FIELDNAMES ...
- (f) CHECK FOR EXISTING FILE NAMES ...

Commentaar bij het scherm

- (a) Voer DATABASE-bestandsnaam in. Gebruik een bestaande database – ZOEK is hier als voorbeeld genomen.
- (b) Voer INDEX-bestandsnaam in. De bestandsnaam mag verschillen van of gelijk zijn aan de bestandsnaam van de DATABASE.
- (c) Voer index zoekveld in. Het betreft hier de veld- of sleuteluitdrukking waarop u uw database geïndexeerd wenst te hebben.
- (d) Als het indexbestand niet bestaat verschijnt de volgende prompt op het scherm
INDEX FILE DOES NOT EXIST. CREATE IT? (Y/N) Y
Voer Y in. Als u een N invoert of < return > aanslaat, verschijnt er een prompt-stip (geen boodschap) en moet u opnieuw met het commando .DO FILEGEN beginnen.
- (e) De boodschap “Creating index file” verschijnt alleen als het indexbestand niet bestaat en gecreëerd moet worden. De boodschap “Getting fieldnames” verschijnt wanneer FILEGEN de veldnamen uit de index van de database heeft verkregen.
- (f) “Check for existing filenames.....” verschijnt terwijl FILEGEN nagaat of er reeds bestandsnamen op de schijfindex staan. Als dat het geval is, brengt FILEGEN prompts op het scherm waarmee u kunt beslissen de bestaande bestandsnamen al dan niet te wissen.

Genereren

- 3. FILEGEN genereert de groep commandobestanden die de ZOEK-database voorzien van de functies “VIEW, ADD, EDIT & PACK”. De gegenereerde programmacode verschijnt regel na regel op het scherm, zodat u kunt zien wat FILEGEN aan het doen is.
- 4. Wanneer de nieuwe programma's gebruiksklaar zijn, meldt FILEGEN het volgende:

TO START “ZOEK” SYSTEM, TYPE THE FOLLOWING

```
.  
.  
.DO ZO-MAIN < return >  
.  
.
```

Opmerking

In het DO ..-MAIN commandoformat, verwijzen de twee letters voor het koppelteken naar de eerste twee letters van de naam van uw database.

8 ... dGEN

Einde FILEGEN

5. U heeft de dBASE II prompt voor zich en kunt een dBASE II-commando invoeren.

GEBUIK VAN PROGRAMMA'S, GECEEERD BINNEN FILEGEN – VOORBEELD

Uitvoering

Telkens wanneer u gebruik wenst te maken van de programma's die door FILEGEN gecreëerd zijn voor de ZOEK-database, gaat u als volgt te werk:

1. Bij de stip van de dBASE II-prompt, voert u in

```
.DO ZO-MAIN < return >
```

om het volgende op uw scherm te krijgen

```
ZOEK MAIN MENU
```

- 0. EXIT
- 1. VIEW
- 2. ADD
- 3. EDIT
- 4. PACK

```
SELECT :2:
```

Records Toevoegen

2. Sla 2 aan om records toe te voegen (APPEND) aan de ZOEK-database, die nu nog leeg is. Op het scherm verschijnt het format voor de gegevensinvoer:

```
ADD ZOEK                                DD/MM/JJ

PERSON      :MW. NICOLE SMIT < return >
AFDELING    :PERSONEELSAFDELING < return >
BEDRIJF     :COMPUSHOP < return >
STRAAT      :
STAD        :
POSTCODE    :
DATUM       :
COMMENTAAR  :
```

```
PRESS < control-W > TO EXIT
```

Vul voor elk veld de informatie in en sla < return > aan. Elke keer wanneer u alle informatie voor een record heeft ingevuld verschijnt er een nieuw scherm voor het volgende record.

Wanneer u klaar bent, houdt u de control-toets ingedrukt en typt u in een leeg record de letter **W** in om terug te keren naar het ZO-MAIN MENU. Sla **0** (nul) aan wanneer u de dBASE II-prompt wenst; **1** om de records in de ZOEK database te bekijken; **3** om ze te wijzigen; **2** om records toe te voegen.

Omdat u de wijzigingsoptie onbenut heeft gelaten zijn er in de database geen records met een wismarkering, en heeft de optie **4, PACK**, nog geen zin.

Records Wijzigen

- Kies optie **3** om de reeds in de database aanwezige records te wijzigen. Het eerste record verschijnt als volgt op het scherm:

EDITZOEK	DD/MM/JJ
PERSOON	:MW. NELLY STEVENS
AFDELING	:PERSONEELSZAKEN
BEDRIJF	:INTETRIM BV
STRAAT	:890 BECKERSTRAAT
STAD	:HOOIK
POSTCODE	:2562 FF
DATUM	:12/2
COMMENTAAR	:PRODUCTIE-MANAGER

COMMAND: (E)DIT (D)ELETE (U)NDELETE (C)ONTINUE (P)OSITION P

De promptregel voor het EDIT scherm bevat:

COMMAND: (E)DIT (D)ELETE (U)NDELETE (C)ONTINUE (P)OSITION

- E- EDIT. Staat u correctie toe van ieder veld in het record behalve het sleutelveld, dat in halve helderheid is afgebeeld. Als er een fout in het sleutelveld staat, moet u het record wissen en gebruik maken van optie 2, ADD, om het hele record opnieuw in te voeren
- D- DELETE. De boodschap "DELETED" verschijnt links van de datum in de kopregel. Wilt u het record verwijderen, dan moet u optie 4, PACK, gebruiken.
- U- UNDELETE. Schrap de "DELETED" boodschap.
- C- CONTINUE. Als u het actueel verschijnende record noch wijzigde, wiste of de wissing ervan ongedaan maakte (UNDELETE), brengt CONTINUE het volgende record op het scherm.

10 ... dGEN

P- POSITION. Brengt een andere "promptregel" voor EDIT op het scherm; en wel als volgt:

EDITZOEK	DD/MM/JJ
PERSOON	:MW. NELLY STEVENS
AFDELING	:PERSONEELSZAKEN
BEDRIJF	:INTETRIM BV
ZELFDE RECORD	STRAAT :890 BECKERSTRAAT
	STAD :HOOIK
	POSTCODE :2562 FF
	DATUM :12/2
	COMMENTAAR :PRODUCTIE-MANAGER

2de EDIT
prompt COMMAND: (D)ISPLAY (F)IND (L)OCATE (C)ONTINUE (S)KIP

Voor de 2e EDIT-promptregel, kiest u optie P in het eerste EDIT scherm.
Deze promptregel bevat:

COMMAND: (D)ISPLAY (F)IND (L)OCATE (C)ONTINUE (S)KIP

D- DISPLAY. Zorgt ervoor dat u meer dan één record tegelijk kunt bekijken en is gebaseerd op de LOCATE uitdrukking. Voer de LOCATE uitdrukking in. Bijvoorbeeld

BEDRIJF = "Compushop"

Voer vervolgens de DISPLAY-reeks in. Bijvoorbeeld

PERSOON + BEDRIJF

F- FIND. Voer de tekst of het sleutelwoord (BEDRIJF) in voor het gewenste record. Bijvoorbeeld

Compushop

L- LOCATE. Gebaseerd op de LOCATE uitdrukking. Voer de LOCATE uitdrukking in. Bijvoorbeeld BEDRIJF="COMPUSHOP". Dit is de basis voor (D)ISPLAY en (C)ONTINUE.

C- CONTINUE. Ga naar het volgende record. Geef aan het einde van het bestand < return > om terug te keren naar de COMMAND-prompt.

S- SKIP. Sla het huidige record over. Geef aan het einde van het bestand < return > om terug te keren naar de COMMAND-prompt.

Bekijken

VIEWZOEK

DD/MM/JJ

4. Kies optie 1 voor een "VIEW" op welk record dan ook in de database. De kop van het scherm vermeldt VIEW i.p.v. EDIT maar overigens is het gelijk aan het EDIT-schermb met de 2e promptregel.

De VIEW-opties werken op dezelfde manier als in het tweede EDIT scherm – het verschil is dat VIEW geen veranderingen van de records toelaat, terwijl EDIT veranderingen toelaat van elk veld met uitzondering van het index-sleutelveld.

Pack

PACKZOEK

DD/MM/JJ

PACK THE ENTIRE FILE? [YES/NO] YES < return >

ZOEK.OLD WILL BE YOUR BACKUP DATA FILE.

COPY TO ZOEK

00004 RECORDS INDEXED

USE

SET ECHO OFF

STRIKE ANY KEY TO CONTINUE.....

5. Kies optie 4 voor een definitieve verwijdering van alle records die in het eerste EDIT scherm een wismarkering kregen.

Wanneer u gevraagd wordt of u het hele bestand wenst te schonen (PACK), moet u YES of NO intypen.

Als u NO intypt, keert u terug naar het Hoofdmenu voor wijzigingen.

Als u YES intypt, wordt er een backup bestand van het Zoek-bestand gecreëerd waarin de oorspronkelijke gegevens bewaard blijven. Het werkbestand van de database wordt vervolgens gePACKt. Sla een willekeurige toets aan om naar het Hoofdmenu terug te keren.

RAPPORTEERPROGRAMMA

FORMGEN

FORMGEN creëert een commandobestand dat uit een bestaande database gegevens put en deze in rapporten opmaakt (zoals met het dBASE II commando REPORT). De output kan geprint worden op het scherm of op een printer. Het volgende korte rapport is gecreëerd uit de ZOEK database.

VACATURES

GEADVERTEERDE BAAN	BEDRIJFSNAAM	DATUM
Bedrijfsleider	E.M.W.S.	7/3
Inkoper	Karl Knudsen BV	8/3
Technisch Directeur	Holland Boorautomaten	10/3
Productieleider	LH Research	5/3
Programmeur	Compushop	3/3
Projectleider	Grootscheeps Europe	14/3
Advertising Manager	Western Digital	25/2
Mediaplanner	Idee & Advies	15/2

OPMERKING

De bovenstaande output had nuttiger kunnen zijn als de database gesorteerd was op COMMENTAAR of BEDRIJF-naam alvorens het rapportcommando te geven.

PROCEDURES

Om een REPORT-programma te creëren en te gebruiken, gaat u als volgt te werk:

Toegang

1. Start FORMGEN direct door een commando vanuit de systeem- of dBASE II-prompt in te voeren:

```
A > DBASE FORMGEN of .DO FORMGEN
```

Zo u wilt kunt u FORMGEN oproepen via het dGEN Main Menu. Kies optie 3.

Prompts

2. FORMGEN geeft onderstaande meldingen weer, en wel één of twee tegelijk. Uw antwoorden die op de ZOEK database van toepassing zijn vindt u direct achter de :

- (a) ENTER DATABASE FILE NAME :ZOEK < return >
- (b) ENTER REPORT FORM
FILENAME :BANEN < return >

ENTER OPTIONS
- (c) LEFT MARGIN < 1 > :10 < return >
LINES/PAGE < 56 > :< return >
PAGE WIDTH < 80 > :55 < return >
- (d) ENTER PAGE HEADING :BAANLIJST < return >
- (e) ARE TOTALS REQUIRED?
(Y/N) :N < return >
SUBTOTALS IN REPORT?
(Y/N) :N < return >
- (f) ENTER COLUMN DESCRIPTORS:
 - 1. WIDTH, CONTENTS :20,COMMENTAAR < return >
HEADING :GEADVERTEERDE BAAN < return >
 - 2. WIDTH, CONTENTS :20,BEDRIJF < return >
HEADING :BEDRIJFSNAAM < return >
 - 3. WIDTH, CONTENTS :5,DATUM < return >
HEADING :DATUM < return >
 - 4. WIDTH, CONTENTS :< return >

Commentaar bij het scherm

- (a) Voer de naam in van het bestand uit de database waarover uw rapport gaat.
- (b) Voer de bestandsnaam voor het REPORT in.
- (c) Voer de opties in voor de Linkermarge, Regels/pagina, en Paginabreedte. Indien de standaardwaarden voldoen, geef < return >. Indien niet, voer de gewenste waarden in en geef < return >.
- (d) Voer de Titel van de Pagina in – d.w.z. een gewenste groep tekens die binnen één regel passen. De titel van de pagina wordt in het midden geplaatst van de paginabreedte die u hierboven gespecificeerd heeft.

14 ... dGEN

- (e) Totalen en subtotalen. Zijn totalen vereist? Subtotalen? Zo ja (Y), creëert FORMGEN de noodzakelijke programmacode voor elke kolom waarop totalen van toepassing zijn. U kunt ieder veld in de database als een sub totaal-veld bepalen. Als de veldnaam die u invoert niet in de index van de database gevonden wordt, verschijnt opnieuw de melding "Enter subtotal field".
- (f) ENTER COLUMN DESCRIPTORS. Deze meldingen zijn voor de specificatie van de kolombreedte, veldnamen, en kolomkoppen (indien gewenst).

Width/Breedte. Indien de door u gespecificeerde kolombreedte kleiner is dan de breedte van het veld in de database wordt de informatie in de betreffende kolom ingekort.

Next column/Volgende kolom. Na elke kolom die u beschrijft, verschijnen een nieuwe kolom en nieuwe prompts voor de invoer van de volgende groep kolombepalingen. Nadat U alle gewenste kolommen beschreven heeft geeft u een < return >.

OPMERKING

U kunt maximaal twaalf kolombepalingen invoeren.

PROCEDURES (Vervolg)

Create/Aanmaken

- 3. FORMGEN begint met het schrijven van het programma dat het rapport creëert waarom wordt verzocht. Tijdens dit proces verschijnt de programmacode op het scherm – te snel voor zorgvuldige lezing, doch langzaam genoeg om te begrijpen wat er aan de gang is.

Output

- 4. Voor een geprinte versie van het rapport – op het scherm of op papier – moet u terugkeren naar dBASE II. Kies de optie 0 (nul) om FORMGEN te verlaten.

Bij de prompt-stip, typt u het commando DO < rapportbestandsnaam >, waarbij < rapportbestandsnaam > de naam betreft die u bij stap 2 heeft ingevoerd. In ons voorbeeld, ziet het commando er als volgt uit:

.DO BANEN < return >

Het programma vraagt of u een output naar het scherm (S) of naar de printer (P) wenst. Voer uw keuze in en het programma loopt onmiddellijk. Wanneer het programma afgelopen is, verschijnt opnieuw de dBASE II prompt.

ADRESLABEL-PROGRAMMA**LABELGEN**

LABELGEN creëert een programma dat zijn gegevens uit een bestaand database-bestand put en ingedeeld als een adreslabel bijeenzet. Hieronder zijn 3 labels afgedrukt door een LABELGEN programma met de ZOEK-database.

1	De heer J. de Jong
2	Afdeling Phono
3	Geluidsmakers BV
4	Stillestraat 14
5	4560 DG DORINGERPLAS
-	
-	
1	Berkam & Kamper
2	Rummelse weg 7
3	3333 MM WACHTEL
4	
5	
-	
-	
1	Mw. Nicole Smit
2	Personeelsafdeling
3	Compushop
4	Bitlaan 8/16
5	3264 KB SCHIJF-AAN-DE-WAL

Een andere toepassing van LABELGEN

De ZOEK-database heeft omvangrijke records bestaande uit 9 velden – méér informatie dan past op één enkel REPORT van 80 of 132 kolommen. Met LABELGEN kunt u een programma maken dat de volledige inhoud van ieder record afdrukt in een passende, gebruiksvriendelijke vorm. Hiertoe past u LABELGEN op de gebruikelijke manier toe, maar maakt u een lijst van de gewenste velden (in iedere volgorde die u maar wenst) en drukt die af op papier en niet op adreslabels.

PROCEDURES

Voor de aanmaak en het gebruik van een adreslabel-programma gaat u als volgt te werk:

Toegang

1. U kunt LABELGEN direct starten door de invoer van een commando vanuit de systeem- of de dBASE II-prompt:

```
A > DBASE LABELGEN of .DO LABELGEN
```

Zo u wilt kunt u LABELGEN ook bereiken vanuit het dGEN MAIN MENU. Kies optie 4.

Invoer-prompts

2. De volgende meldingen (prompts) verschijnen wanneer u LABELGEN oproept; uw antwoorden die betrekking hebben op de ZOEK database volgen op de dubbele punten.

- (a) ENTER DATABASE FILENAME: ZOEK < return >
- (b) ENTER LABEL FILENAME: BAANADR < return >
- (c) ENTER LABEL LINES (EXAMPLE):
 - 1:NAME
 - 2:[COMPANY] (Afdrukken wanneer niet leeg)
 - 3:ADDRESS
 - 4:TRIM(CITY) + " , " + STATE + " " + ZIP
- (d) ENTER LABEL LINES:
 - 1:PERSON < return >
 - 2:[AFDELING] < return >
 - 3:BEDRIJF < return >
 - 4:STRAAT < return >
 - 5:TRIM(POSTCODE) + " " + STAD < return >
- (e) 6: < return >

AANTEKENINGEN

- (a) Voer de bestandsnaam van de database in. Als LABELGEN de door u gespecificeerde database niet vindt, wordt het scherm gewist en verschijnt de volgende boodschap: Sla een willekeurige toets aan om verder te gaan... Wanneer u een toets aanslaat verschijnt de dBASE II prompt. Zorg er voor dat u de juiste naam van de database kent en dat u de juiste drive opgeeft als de database zich niet op de standaard-drive bevindt.

- (b) Voer de LABEL-bestandsnaam in. De LABEL-bestandsnaam is de naam van een commandobestand voor het format van het adreslabel in kwestie. Het zal alleen labels creëren op basis van gegevens uit de database die op de voorgaande regel is opgegeven.
- (c) Het voorbeeld toont u hoe u veldnamen kunt invoeren voor een 4-regelig label. Dit voorbeeld verschijnt altijd op deze manier op uw scherm, boven de eigenlijke label-prompts.
- (d) Voer LABEL-regels in. Dit is de plaats waar u feitelijk de informatie invoert die LABELGEN nodig heeft voor de aanmaak van het “label-genererende” programma. Iedere veldnaam die door u wordt opgegeven tussen vierkante haken, wordt slechts dan op een adreslabel afgedrukt als het betreffende veld niet leeg is. Hier maakt u gebruik van de ZOEK-database voor het creëren van de adreslabels.
- (e) Nadat u alle labelregels heeft ingevoerd, slaat u < return > aan.

Aanmaken

- 3. LABELGEN genereert het commandobestand dat de verzochte adresstroken creëert. De commandoregels verschijnen op het scherm terwijl het programma gegenereerd wordt.

Wanneer het nieuwe programma gebruiksklaar is, zal LABELGEN u terugbrengen, ofwel naar de dBASE II-prompt (nl. als u LABELGEN aanriep vanuit dBASE), ofwel naar het dGEN Main Menu (nl. als u LABELGEN binnenkwam vanuit het dGEN Main Menu).

Output

- 4. Om al de adreslabels af te drukken – op het scherm of op papier – moet u zich bij de dBASE II prompt bevinden.

Bij deze prompt-stip typt u DO < label-bestandsnaam > , waarbij < label-bestandsnaam > de naam is voor het label-programma. In ons voorbeeld ziet het commando er als volgt uit.

.DO BAANADR < return >

- 5. Op het scherm verschijnt “MAILING LABELS” met een prompt zoals hieronder getoond wordt. Voer S in voor scherm; P voor printer, of < return > om het commando te annuleren en terug te keren naar de dBASE II-prompt.

ZOEK MAILING LABELS DD/MM/JJ
OUTPUT TO THE SCREEN OR PRINTER? [S/P] P

HET AANMAKEN VAN MAIN MENUS (HOOFDMENU'S)

MENUGEN

MENUGEN creëert programma-menu's. Zo werd bijvoorbeeld het dGEN Main Menu gecreëerd door MENUGEN. U kunt MENUGEN gebruiken voor bijvoorbeeld de onderstaande menu's.

Menu-Titel	HOOFDMENU ZOEK
Opties	0. EXIT 1. BEKIJK, WIJZIG OF WIS RECORDS 2. AFDrukKEN RAPPORT BANEN 3. AFDrukKEN LABELS BAANADR

Meldregel KIES ::

Menu-Titel	HOOFDMENU SPELLETJES
Opties	0. EXIT 1. EZELTJE PRIK 2. PAKMAN 3. KERKERS EN DRAKEN

Meldregel KIES ::

BELANGRIJK

Na het maken van een menu kunt u het op het scherm brengen door het programma met een DO-commando op te roepen. Als u dan echter een optie kiest, zal het scherm knipperen en weer onveranderd terugkeren (opnieuw het hoofdmenu afbeelden).

Om het menu te laten werken heeft u twee dingen nodig – programma's die uitvoeren wat uw opties beloven, en kleine aanpassingen van het door MENUGEN gecreëerde programma. Om te vernemen wat u hier moet doen, zie 'Alleen Voor Programmeurs' (pag. 21).

BENODIGDHEDEN

Wanneer u voor het maken van een menu MENUGEN gebruikt, is het bestaan van een database niet noodzakelijk. U heeft zelfs geen werkende programma's nodig om de opties te ondersteunen. Het enige dat u nodig heeft is de lijst met functies (opties) die het menu te bieden heeft.

PROCEDURES

Voor het maken en gebruiken van hoofdmenu's voor wat voor dBase-programma's dan ook, doet u het volgende:

Toegang

1. De snelste manier om MENUGEN te starten geschiedt door de invoer van

```
A > DBASE MENUGEN      - vanuit de systeemprompt of
.DO MENUGEN            - vanuit de dBASE II-prompt
```

Het kan zijn dat u liever eerst het dGEN-Hoofdmenu aanroept en dan via optie 1 MENUGEN laat starten; om het Hoofdmenu aan te roepen, voert u in:

```
A > DBASE DGEN      of
.DO DGEN
```

Invoer Prompts

2. MENUGEN brengt de prompts (meldingen) op het scherm die hieronder worden getoond; de eerste twee antwoorden (achter de dubbele punten) creëren het programma dat kan worden ingezet om het hoofdmenu voor de ZOEK database op het scherm te brengen.

```
(a) ENTER PROGRAM NAME :VINDBAAN < return >
(b) ENTER MENU HEADING :HOOFDMENU ZOEKBAAN < return >
(c) ENTER MENU OPTIONS :
    0:  EXIT
    1:  BEKIJK, WIJZIG OF WIS RECORDS < return >
    2:  PRINT BANEN RAPPORT < return >
    3:  PRINT BAANADR LABELS
    4:  < return >
```

```
(d) COMMAND: (E)XIT (R)EDO (S)AVE
```

Commentaar bij het scherm

- (a) Gebruik de PROGRAM naam om het hoofdmenu op te roepen wanneer u gereed bent voor het gebruik van de opties;
- (b) De MENU-Titel betreft de meer uitgebreide titel die boven aan het nieuwe hoofdmenu verschijnt. De antwoorden in het hier bovenstaande voorbeeld stellen u in staat het Hoofdmenu Zoekbaan op te roepen door het invoeren van het commando .DO VINDBAAN < return >;

20 ... dGEN

- (c) Voer de menu-opties in. Optie 0 (nul) betreft altijd "EXIT" – de optie die u doet terugkeren naar de promptstip van dBASE II – tenzij u het MENUGEN-programma wijzigt. Voer optie 1 in, geef vervolgens < return > om de volgende optie in te voeren, enz. U kunt doorgaan tot 14 opties. Wanneer u klaar bent, geeft u een < return > .
- (d) COMMAND (E)XIT (R)EDO (S)AVE

Waarin:

- (E)XIT – terugkeert naar de dBASE II prompt of naar dGEN, zonder een menuprogramma te creëren.
- (R)EDO – alle prompts herhaalt, vanaf het begin (dus ook programma-naam en menu-titel). Om opnieuw te kunnen beginnen.
- (S)AVE – het genereren van uw menuprogramma start, uitgaande van uw antwoorden op de prompts.

PROCEDURES (vervolg)

Programma genereren

- Als u (S)AVE kiest, genereert MENUGEN het programma dat u in staat stelt het nieuwe Hoofdmenu op te vragen. U ziet op het scherm de regels met de programmacode die in de maak is.
- Wanneer uw nieuwe Hoofdmenu-programma klaar is, zal MENUGEN terugkeren naar de dBASE II-prompt (of naar het dGEN Hoofdmenu, als u MENUGEN van daaruit heeft gestart).

Het menu oproepen

- Om uw nieuwe Hoofdmenu op het scherm te brengen, typt u DO < programmaam > . In dit geval:
.DO VINDBAAN < RETURN >

ALLEEN VOOR PROGRAMMEURS

HET AANMAKEN VAN MENU'S

De Hoofdmenu-programma's die u met MENUGEN maakt werken niet voordat u het programma aanpast. Gebruik het dBASE II MODIFY COMMAND om het programma op te roepen.

1. Ga naar de .DO CASE instructies. Merk op dat elke optie van het Hoofdmenu genoteerd staat als een CASE.
2. U voegt nu een nieuwe regel in bij ieder CASE-commando en voert DO <programma-bestandsnaam> in. Bijvoorbeeld

DO BANEN

3. Zolang u zich binnen MODIFY COMMAND bevindt, kunt u desgewenst
 - (a) uw naam invoeren in de titel van het programma en
 - (b) elke gewenste verandering aanbrengen in de tekst van de opties zoals ze worden afgebeeld in het Hoofdmenu. U zou bijvoorbeeld de tekst van de 0 (nul) optie kunnen veranderen van "EXIT" in "EINDE" of "TERUG NAAR DE dBASE-PROMPT".

dGEN

FILEGEN gebruikt het volgende format om de programma's te benoemen die u creëert onder FILEGEN.

DD-PROGRAM.EXT

Waarin

DD=de eerste 2 letters van de naam van de database; bijv. ZO voor de ZOEK-database;

PROGRAM=MAIN, FRAME, SOME, GETS, EDIT, PACK, POSN en LOCAT; en

EXT=de uitbreiding van het programma: .CMD voor CP/M-80 programma's, en .PRG voor MS-DOS of CP/M86-programma's.

OPMERKING

Gereserveerde woorden vindt u op pagina 2.



Samenvatting van de
wijzigingen t.o.v. Versie 2.4

ASHTON · TATE ■TM

© 1985 Ashton-Tate
Reproduction by any method is strictly prohibited

UITBREIDINGEN EN NIEUWE KENMERKEN

@ < coördinaten > GET < variabele > PICTURE "AAAAA" – Invoeren van spaties

Versie 2.4 accepteerde de invoer van ieder letterteken ('alfa') indien de PICTURE-clausule de letter 'A' bevatte. Versie 2.41 accepteert nu ook spaties.

CLEAR – Effect op formatbestanden

In versie 2.4 betrof het CLEAR-commando het vrijgeven van alle actieve geheugenvariabelen en de sluiting van alle open database-bestanden. Het CLEAR-commando sluit **nu** ook de formatbestanden. SET FORMAT TO, zonder argument, sluit nog steeds formatbestanden zonder iets anders te verstoren.

CREATE – Afbreken met de < escape > toets

Wanneer men **vroeger** als antwoord na ENTER FILENAME de < escape > -toets aansloeg kon dBASE II hierdoor vastlopen.

dBASE II accepteert nu probleemloos < escape > als antwoord op de ENTER FILENAME prompt.

CREATE < bestand > FROM < structuur-bestand > – Beperking

In **vroegere** versies kon men op deze manier maximaal 14 opeenvolgende bestanden creëren.

Nu is er geen beperking meer.

INSTALL – Keuze van scheidingstekens voor variabelen in full screen modus

Nu mag ieder teken gespecificeerd worden als een scheidingsteken. Hoewel de standaard-scheidingstekens nog steeds dubbele punten zijn, mag nu elk tekenpaar (zoals < > of []) gekozen worden.

2 ... dBASE

INS- en DEL-toetsen op de IBM-PC – Nu werkzaam

De INSert- en DElete-toetsen op de IBM-PC werken **nu** in dBASE II op dezelfde wijze als in PC-DOS.

- * INS schakelt de invoegmodus aan en uit
- * DEL wist het karakter onder de cursor.

JOIN, SORT, en TOTAL – Afbreken met de < escape > toets

Als de < escape >-toets wordt aangeslagen om de commando's JOIN, SORT, of TOTAL af te breken, wordt **nu** het nieuwe bestand dat in de maak was, automatisch gesloten.

Vroeger bleef het open en verscheen de boodschap FILE IS CURRENTLY OPEN wanneer men een poging ondernam het te wissen met < delete >.

REPORT FORM < structuur-bestand > – Fout in structuurbestand

Wanneer **vroeger** het REPORT-commando een fout ontdekte in een in een bestand opgeslagen structuur, verscheen de boodschap SYNTAX ERROR en ging het systeem over op de foutcorrectie-dialoog. Iedere poging om de syntaxis te corrigeren en terug te keren naar het (in de maak zijnde) rapport, resulteerde in de boodschap: FILE IS CURRENTLY OPEN.

Nu stopt REPORT wanneer het een fout tegenkomt, brengt een nieuw bericht, SYNTAX ERROR IN FORMAT SPECIFICATION, op het scherm en breekt de uitvoering van het REPORT-commando af. Als het REPORT-commando is gegeven vanuit de promptstip keert het systeem daarnaar terug. Als REPORT is gegeven vanuit een commandobestand, zal de uitvoering van commando's hervat worden vanaf de coderegel die onmiddellijk volgt op het REPORT-commando.

RESTORE FROM < geheugenbestandsnaam > ADDITIVE – Geheugenvariabelen met dezelfde naam

Als er een geheugenvariabele gedefinieerd is onder dezelfde naam als een variabele die is opgeslagen in een geheugenbestand zal het RESTORE-commando met de ADDITIVE-parameter **nu** tot effect hebben dat de geheugenvariabelen die zijn 'restored', de reeds bestaande overschrijven.

SAVE TO < geheugenbestandsnaam > ALL EXCEPT < masker > – Nieuwe parameter

Accepteerde **vroeger** het SAVE TO commando slechts de parameter ALL LIKE < masker >, **nu** aanvaardt het ook ALL EXCEPT < masker >.

ATTENTIE!

@ < coördinaten > GET < logische variabele > ...READ – Waarde onveranderd

Vroeger gold, dat het aanslaan van < return > een < logische variabele > met een waarheidswaarde 'waar' (.T.) veranderde in niet waar (.F.)

Nu blijft de waarde dezelfde zolang de operator deze niet doelbewust verandert.

APPEND FROM < bestandsnaam database > – Afbreken met < escape >

Wanneer in **vorige** versies een commando APPEND FROM werd afgebroken met de < escape > -toets, raakte het USE-bestand in de war.

Nu breekt de < escape > -toets APPEND op een ordelijke wijze af (na elke 100 records), en blijft de integriteit van het USE-bestand gehandhaafd.

APPEND FROM < bestandsnaam database > – 1000 tekens-records

Wanneer **vroeger** het USE- en het FROM-bestand dezelfde structuur met 1000 tekens per record hadden, voegde APPEND FROM lege records toe aan het USE-bestand.

Nu worden alle gegevens op correcte wijze aan het USE-bestand toegevoegd.

APPEND FROM < bestandsnaam tekst > .TXT – Weglating van de parameter 'SDF'

Wanneer u **vroeger** probeerde toe te voegen (APPEND) vanuit (FROM) een tekstbestand met een commando zonder SDF-bepaling, gaf dBASE II het bericht NOT A dBASE II DATABASE. Als u dan de fout verbeterde in de fout-correctie-dialoog, gaf dBASE II het bericht FILE IS CURRENTLY OPEN.

Wanneer u **nu** de correctie maakt, verwerkt dBASE II het gecorrigeerde commando.

4 ... dBASE

Commando-bestanden – Vreemde word processors en commando-inkorting

Sommige **oudere** versies herkenden het end-of-file-teken niet in commandobestanden die opgemaakt waren met andere word processors of tekstverwerkers dan MODIFY COMMAND. Dit verhinderde de verwerking van het laatste 128-byte blok code.

Nu wordt alle code verwerkt, zelfs als de totale omvang van het bestand kleiner is dan 128 bytes.

CREATE [< bestandsnaam >] – In MS(PC)-DOS, versies 2.01 en 2.1

Vroegere versies reageerden op het CREATE-commando met de boodschap END OF FILE FOUND UNEXPECTEDLY. In deze DOS-versies mogen bestandsnamen internationale tekens bevatten.

Nu ondersteunt dBASE II deze mogelijkheid.

INDEX – Bijwerkingstijd

Gedurende commando's zoals APPEND, EDIT en READ neemt het bijwerken van de index **nu** geen ongewoon lange tijd in beslag.

INSERT – Met een format-bestand

Wanneer men **vroeger** het INSERT-commando met SET FORMAT TO < naam formatbestand > gebruikte, verscheen na de voltooiing van de data-invoer de foutmelding RECORD OUT OF RANGE.

Nu verschijnt er geen foutmelding meer en keert dBASE II gewoon terug naar de promptstip in het geval van een interactief programma, of naar de volgende regel code in het geval van een commandobestand.

JOIN TO < nieuwe bestandsnaam > FOR P. < sleutel > = S. < sleutel > – “Duplicaatvelden”

In **vroegere** versies ontstonden bij samenvoeging (JOIN) van twee bestanden met gelijke veldnamen duplicaatvelden in het nieuwe bestand. Wanneer dan het nieuwe bestand de 32 velden overschreed, kon het systeem vastlopen.

Nu creëert JOIN geen duplicaatvelden, en omvat het nieuwe bestand alleen de eerste 32 velden.

Logische velden – in een grote database-structuur

Wanneer **vroeger** een database structuur de grens van 1000 tekens naderde en er in de structuur een logisch veld volgde op een numeriek veld, verstoorde de uitvoering van het LIST-commando de bestandsstructuur.

Nu is een dergelijke structuur aanvaardbaar.

QUIT TO < lijst commando's > – In 16-bit systemen

QUIT TO is uitvoerbaar gemaakt voor CP/M-86, versie 1.1, en MS-DOS, versie 2.0. Om terug te keren naar dBASE II, moet dBASE opgenomen worden in de < lijst commando's >.

Voor MS(PC)-DOS, versie 2.0 geldt dat de besturing terugkeert naar de plaats in dBASE II waar het QUIT TO werd uitgevoerd. In een commandobestand keert het commando QUIT TO terug naar de volgende regel code en wordt de uitvoering van de commando's hervat. Geheugenvariabelen en alle parameters blijven ongestoord. Echter, alle databestanden en de op hun betrokken index- en formatbestanden zijn gesloten ter vermindering van elke interactie met bestandsaanroep in andere programma's. Als dBASE in de < lijst commando's > opgenomen is, wordt een nieuwe sessie dBASE II gestart die "genest" wordt binnen de oude. De oude wordt dan weer opnieuw ingevoerd na de voltooiing van de < lijst commando's >. De DOS-commando-interpret, COMMAND.COM, moet zowel aanwezig zijn op de boot-drive als op de drive waarin dBASE II zich bevindt.

REPLACE < numerieke veldnaam > WITH < getal > – Groter dan 50 cijfers

Wanneer **vroeger** het < getal > meer dan 50 cijfers in beslag nam, werd het USE-bestand vernietigd.

Nu wordt het < getal > tot 34 cijfers ingekort die zonder problemen in het veld worden vervangen. Slechts de eerste tien cijfers zijn significant, de rest zijn nullen en het hele getal wordt geheel rechts in het veld geplaatst.

SKIP < negatieve geheugenvar > – Negatieve < geheugenvar > herkend

Vroeger las SKIP < negatieve geheugenvar > als een positief getal.

Het herkent **nu** het minteken wel en springt het opgegeven aantal records terug. 'SKIP – < negatieve geheugenvar >' springt in het bestand voorwaarts.

6 ... dBASE

SKIP VAL (" - 1") – Negatief getal herkend

SKIP las **vroeger** VAL (< negatief getal >) als een positief getal.

Het herkent **nu** het minteken wel en springt het opgegeven aantal records terugwaarts.

Systeemdatum – Formatverandering

Zoals in vroegere versies biedt het INSTALL-programma de keuze van dBASE II-systeem-datumsformats: het Amerikaanse (mm/dd/jj) of Europese (dd/mm/jj). **Vroeger** kon men slechts éénmaal zo'n keuze doen.

Nu kunnen de formats veranderd worden wanneer u maar wilt door het INSTALL-programma opnieuw uit te voeren.

TOTAL ON < sleutelveld > TO < bestandsnaam > – Eerder bestaande TO-database

Vroeger moesten de overeenkomende numerieke velden van de beide databases een identieke omvang hebben, wilde men onjuiste resultaten in de totalen vermijden.

Nu kunt u van tevoren aan de TO-database een grotere veldbreedte toewijzen dan de USE-database. Als de numerieke velden in de TO-database groot genoeg zijn om de uiteindelijke totalen te bevatten zullen de totalen correct bepaald worden.

UPDATE met REPLACE – Numerieke overflow

Wanneer men **nu** een UPDATE uitvoert met een REPLACE van een numeriek veld naar een kleiner veld in de primaire database, wordt het getal in het numerieke veld niet beknot maar neemt een serie asterisken (*****) de plaats van de getallen in om de numerieke overflow aan te geven.

WRAP-AROUND op het IBM-PC scherm – Verlies van een teken

Wanneer **vroeger** een output de schermbreedte van 80 tekens overschreed en overging naar de volgende regel ("wrap around") bleek het 81e teken te ontbreken.

Nu verschijnen alle tekens op het scherm.

NAGEKOMEN DOCUMENTATIE**@ < coördinaten > GET < numerieke variabele > PICTURE “ < format > ” – Overflow**

Als de breedte van het < format > kleiner is dan die van de < numerieke variabele >, zoals in het voorbeeld

```
STORE 123.45 TO getal
@ 5,5 GET getal PICTURE "99.99"
READ
```

verschijnt het volgende op het scherm

```
**34
```

Asterisken geven een numerieke overflow aan.

Als het getal een fractie is moet het format groot genoeg zijn om de decimale punt en de daaraan voorafgaande nul te bevatten.

```
STORE .12 TO decimaal
@ 5,5 GET decimaal PICTURE "#.#"
```

@ < coördinaten > GET < numeriek veld > PICTURE “ < format > ” – Overschrijven

Als het < numerieke veld > ruimte voor decimalen heeft, dan is de invoer onjuist wanneer men een poging onderneemt achter de decimale punt te schrijven. Dit is te wijten aan de aanwezigheid in het veld van een vaste decimale punt die niet, zoals in een geheugenvariabele, kan worden overschreden. Gebruik de geheugenvar voor invoer en REPLACE de data van de geheugenvar naar een veld van de database.

@ < coördinaten > SAY < uitdrukking > USING < format > – Expansie

dBASE II expandeert een tekenreeks (substring) in een USING-zin niet zoals in versie 2.3. Bijvoorbeeld

```
STORE '120883' TO mdatum
@ 5,5 SAY mdatum USING '99/99/99'
```

Het resultaat hiervan is '12/88/ '. Om dit te omzeilen, moet u de substring-functie (\$) gebruiken.

```
@ 5,5 SAY $(mdate,1,2) + '/' + $(mdate,3,2) + '/' + $(mdate,5,2)
```

Nu verschijnt er 12/08/83.

8 ... dBASE

DELETE FILE < parameter > – Geldige en ongeldige syntaxis

LIKE < masker > en EXCEPT < masker > zijn geen geldige syntaxis voor < parameter >. De enige geldige syntaxis voor < parameter > is een bestandsnaam die een ? als een neutraal teken (wildcard) mag bevatten. * is geen geldig neutraal teken in < parameter >. Bijvoorbeeld

```
DELETE FILE TE??.DBF
```

wist alle vier-letterige bestandsnamen die beginnen met TE en de .DBF-uitbreiding hebben. Daarentegen resulteert

```
DELETE FILE TE*.DBF
```

in het bericht FILE NOT FOUND.

FUNCTIETOETSEN – De daaruit voortkomende commando's zijn niet te onderbreken

Wanneer commando's worden uitgevaardigd met de functietoetsen zal het aanslaan van de < escape >-toets het proces niet doen stoppen als de commandoreeks van de functietoets een puntkomma bevat. Wanneer met diverse functietoetsen meerdere commando's worden uitgevoerd, zullen ze alle hun uitvoering voltooien voordat de besturing terugkeert naar de promptstip of het commandobestand.

De puntkomma wordt in de commandoreeks van een functietoets als automatische < return > gebruikt om de uitvoering te starten. Als een commando onderbreekbaar moet zijn, mag het niet aan een functietoets met puntkomma toegewezen worden. Commando's kunnen slechts afgebroken worden als men de < return > op het toetsenbord invoert.

GO/GOTO < parameter > – GO naar gewiste records

De GO/GOTO commando's kunnen naar ieder record gaan, ongeacht zijn wismarkering. Zelfs met SET DELETED ON, als het eerste record van een bestand een wismarkering heeft, brengt GO TOP u nog steeds naar dat record. Een voorwaardelijk commando kan gebruikt worden om een wismarkering te onderkennen.

```
IF *  
    SKIP  
ELSE  
    DISPLAY  
ENDIF
```

INSERT BLANK – Met een open indexbestand

Wanneer het commando INSERT BLANK bij een open indexbestand gebruikt wordt brengt het u in het ingelaste record, in de "full screen" modus. Om een leeg record in een geïndexeerd bestand in te lassen zonder in de full screen mode te geraken, gebruikt u het APPEND BLANK-commando.

LOCATE < omvang > FOR < logische uitdrukking > – 'WHILE' niet geldig

Het gebruik van WHILE in plaats van FOR is een ongeldige syntaxis en zal een foutmelding produceren.

MODIFY STRUCTURE – Een bekende fout in de PC-DOS versie

Deze commandoreeks maakt dat er wartaal op het scherm verschijnt en dat de computer vastloopt.

```
USE < bestandsnaam >
MODIFY STRUCTURE
< control-C >
< control-C >
< control-N >
< control-T >
< control-Q of W >
MODIFY STRUCTURE
< control-C >
< control-C >
< control-N >
```

De uitvoering van een WAIT-commando voorafgaande aan het tweede MODIFY STRUCTURE voorkomt dit probleem geheel.

PEEK [< adres >] – Bekende fout

De functie PEEK() is niet werkzaam van 1000 hex (4096 decimaal) tot 7A00 hex (31232 decimaal). Hoewel hij een getal levert wanneer binnen dit gebied gebruikt, is dit getal niet nauwkeurig.

RELATIEF ADRESSEREN OP DE PRINTER – Niet geïmplementeerd

Het gebruik van het dollarteken (\$) in de parameter < coördinaten > van het @...SAY-commando wordt niet bekrachtigd wanneer SET FORMAT TO PRINT in werking is gesteld. "Relatief adresseren" kan alleen op het scherm.

REPLACE P. < veld > WITH S. < veld > – Kiezen van geschikt gebied

Een REPLACE-commando vervangt slechts velden in het database-bestand dat open is binnen het huidige geSELECTeerde gebied. Bijvoorbeeld, dit werkt:

10 ... dBASE

```
SELECT PRIMARY  
REPLACE P. < veld > WITH S. < veld >
```

Maar dit werkt niet:

```
SELECT PRIMARY  
REPLACE S. < veld > WITH P. < veld >
```

SET COLOR TO < n1 >, < n2 > – Weglating van ‘< n1 >,’

Versie 2.4 stond de weglating toe van de parameter < n1 > en verzorgde de uitvoer SET COLOR TO < n2 >. Deze commandovorm werd nooit als geldig beschouwd, noch was het de bedoeling dat hij zo gebruikt werd. Versie 2.41 verandert de eigenschappen van het scherm niet, tenzij het commando beide parameters < n1 > en < n2 > bevat, hoewel het ontbreken ervan geen syntax-error veroorzaakt.

SET LINKAGE ON – Onjuiste syntaxis

Men schakele het verbindingscommando (LINKAGE) op de juiste wijze in voordat men database-bestanden opent. Zijn de bestanden reeds geopend en wordt daarna SET LINKAGE ON ingevoerd, dan wordt er geen verbinding tot stand gebracht. Commando's als LIST, die normaal stap-voor-stap door het bestand gaan, zullen onvoorspelbaar functioneren.

SORT ON < sleutel > TO < naam bestand na sorteren > – Na APPEND:

Tussen het toevoegen (APPEND) van records en het SORTeren van een bestand moet het gesloten worden. Dit bewerkstelligt men op een eenvoudige manier met het USE-commando, door hetzelfde, reeds in gebruik zijnde, bestand te vermelden:

```
USE BESTAND2  
APPEND  
USE BESTAND2  
SORT ON < sleutelveld > TO < naam bestand na sorteren >
```

STORE – Reeksen groter dan 254 tekens

Met STORE < reeks1 > + < reeks2 > TO < reeks3 > kunt u in < reeks3 > tot 254 tekens aaneenschakelen. Wanneer < reeks3 > de limiet van 254 tekens overschrijdt verschijnt de melding SYNTAX ERROR.

Bij een fout in het opslaan (STORE) van een reeks met 254 tekens in een < variabele >, zal uitbreiding van de lengte van de reeks tijdens het corrigeren van de fout tot een SYNTAX ERROR leiden. Om dit te

vermijden moet de fout gewist worden voordat men tot correctie overgaat.

STORE < uitdrukking > TO < geheugenvar > – Veranderen van het type van de < geheugenvar >

Men kan noch de STR() noch de VAL() functie in de < uitdrukking > gebruiken om het type van een bepaalde geheugenvariabele te veranderen door het opnieuw in zichzelf op te slaan. Dit zal altijd in 0 resulteren:

```
STORE "123" TO tkn
STORE VAL(kar) TO tkn
```

of

```
STORE 123 TO getal
STORE STR(getal,3) TO getal
```

Om het type van een geheugenvariabele te veranderen, moet u eerst met de functie VAL() of STR() een nieuwe variabele creëren en die vervolgens in de eerste variabele opslaan:

```
STORE "123" TO tkn
STORE VAL(tkn) TO tkn2
STORE tkn2 TO tkn
```

TOTAL ON < sleutelveld > TO < bestandsnaam > – Deelreeks niet geldig

De deelreeksfunctie mag niet gebruikt worden in de parameter < sleutelveld >. Als er een TOTAL op een deelreeks ("substring") nodig is, moet u die vervangen door de deelreeks van het sleutelveld, en het nieuwe deelreeks-veld als de < sleutelveld >-parameter gebruiken.

TOTAL ON < sleutelveld > TO < bestandsnaam > – Velden groot genoeg voor totalen

Als de TO-database niet bestaat, moet de USE-database beschikken over een numeriek veld met een omvang die de uiteindelijke totalen kan opnemen. De structuur wordt niet uitgebreid wanneer men haar kopieert in de TO database en zal onjuiste totalen bevatten als zij te weinig plaats bevat.

Een verandering van de veldlengte in de USE-database kan vermeden worden door een TO-database met grotere structuur te creëren.

Index

INDEX

INDEX...2

INDEX

Met deze index kunt u de gewenste informatie snel en gemakkelijk terug vinden. De meeste items staan onder meerdere trefwoorden vermeld. De **vetgedrukte** paginanummers zijn de beste verwijzing voor een bepaald item: op die bladzijde vindt u de meest volledige uitleg. – De Index heeft betrekking op de pagina's dBASE...1 tot en met 148...WAIT.

- ! formatteken, 38
- ! hoofdletter-functie, **14**
- " scheidingstekens, 86
- # formatteken, 38
- # recordnummer-functie, 10
- # vergelijkings-operator, 17
- \$ deelreeks-functie, 11
- \$ formatteken, 38
- & macro-substitutie, **19**
- ' scheidingstekens, 86
- () haakjes voor rekenkundige prioriteit, 16, 17
- * commentaar; opmerkingen, 8, 99
- * formatteken, 38
- * gewist record-functie, 13
- * vermenigvuldiging, 16
- + optelling, 16
- + reeksaaneenschakeling, 18
- aftrekking, 16
- reeksaaneenschakeling, 18
- . dBASE-prompt, 1
- .AND. Boleaans "en", 17
- .CMD commandobestanden, 7
- .DBF databasebestanden, 5
- .FMT formatbestanden, 8
- .FRM rapportformulierbestanden, 7
- .MEM geheugenbestanden, 6
- .NDX indexbestanden, 8
- .NOT. Booleaans "niet", 17
- .OR. Booleaans "of", 17
- .PRG commandobestanden, 7
- .TXT tekstbestand 8, 20
- / deling, 16
- / scheidingstekens voor datums, 1
- 8080, 8086, 8088 microprocessors, 4
- 9 formatteken, 38
- < kleiner dan, 17
- < = kleiner dan of gelijk aan, 17
- < > niet gelijk aan, 17
- = gelijk aan, 17
- > groter dan, 17
- > = groter dan of gelijk aan, 17
- ? commando, interactief, 22, **33**
- ?? commando, **33**
- @ commando, **35**
- @ weergave-commando, 22
- @ en scherm of printer, 38
- @ en READ, 103
- @ opdracht, 35
- @ subreeks-functie, 13

- A -

A – formatteken, 38
 aanroepen van dBASE II, 1
 ACCEPT, 40
 commandobestands-commando, 22
 en INPUT, 86
 ADDITIVE-zin en RESTORE, 126
 afkortingen voor commando's en
 sleutelwoorden, 29
 ALL – definitie, 27
 ALL LIKE en SAVE, 128
 ALTERNATE parameter, 131
 annuleren van een commando, 49
 apparaatcommando's, 23
 APPEND: 41–47, 97
 cursorbesturing, 24
 FROM, voorbeeld, 46
 met INDEX, 147
 en andere programmatuur dan
 dBASE, 20
 SDF-bestanden, 22
 toevoegen van gegevens, 21
 APPEND BLANK, 41
 ASCII sorteervolgorde, 138

- B -

backspace, zie spatie-terugtoets
 BASIC, koppeling aan, 20
 BEFORE-zin, 88
 BELL, 39, 132
 bepalingen:
 BLANK, 41
 DELIMITED, 41
 FIELDS, 143
 FOR, 28, 141
 NEXT, 94
 OTHERWISE, 68, 69
 PICTURE, 37
 SDF, 41, 53
 STRUCTURE, 53
 USING, 37
 WHILE, 28, 143
 bereik – definitie, 27
 bestand – definitie, 5, 27
 bestand-functie FILE, 15
 bestanden, commando's voor

 bewerking van, 22
 bestandsnaam (-namen):
 wijzigen, 111
 definitie, 27
 uitbreiding:
 .CMD, 7
 .DBF, 5, 111
 .FMT, 8
 .FRM, 7
 .MEM, 6
 .NDX, 8
 .PRG, 7
 .TXT, 8
 uitbreidingen, 5
 RENAME, 111
 bestand(en):
 creëren van, 20
 dBASE-bestanden, 5
 FORM, 115
 LIST, 93
 besturingssysteem, 4
 besturingstekens, 3
 bewerking(en), 16
 logische, 16, 17
 PACK, 63
 RECALL, 63
 reeks-, 16, 18
 vergelijkings-, 16, 17
 rekenkundige, 16, 17
 BLANK:
 in APPEND-commando, 41
 in INSERT-commando, 88
 Booleaans AND, 17
 Booleaans NOT, 17
 Booleaans OR, 17
 BOTTOM, GOTO –, 78
 BROWSE, 21, 22, 48
 cursorbesturing in, 25
 gegevens wijzigen, 21, 48
 verschuiven van venster, 25
 BROWSE FIELDS, 48

- C -

CALL, 32
 CANCEL, 23, 49
 CARRY, 88
 CARRY-parameter, 133

4 ... INDEX

- CASE, DO -, 68
- CHANGE, 21, **50**
- CHR:
 - getal-naar-teken-functie, **14**
- CLEAR, 35, **51**
- CLEAR GETS, 36, **51**
- COLON, 39, 132
- commandobestand, 7
 - commando's, 23
 - en DO-commando, 67
 - genest, 7
 - wijzigen van, 97
- commando's, zie Bijlage A
- commando's, beschrijving en gebruik van
 - *, 99
 - ?, 22, **33**
 - ??, 33
 - @, 35
 - weergave van gegevens, 22
 - en READ, 103
 - wijzigen van scherm, 39
- afgekorte commando's, 29
- ACCEPT, **40**
 - commandobestand-commando, 22
 - en INPUT, 88
- APPEND, **41-43**
 - bestandsbewerking, 22
 - en COPY, 20
 - en INDEX, 147
 - en INSERT, 88
 - gegevens toevoegen, 21
- APPEND BLANK, 41
- BELL, 39
- bestandsbewerking, 21
- BROWSE, 21, 22, **48**
- CANCEL, 23, **49**
- CARRY, 88
- CHANGE, 21, **50**
 - commando's, verschillende types, **21**
- CLEAR, 35, **51**
- CLEAR GETS, 36, 51
- COLON, 39
- commandobestandscommando's, 23
- COMMAND, MODIFY -, 97
- CONTINUE:
 - en LOCATE, 52, 94
 - plaatsingscommando, 21
- COPY, **53-57**
 - creëren van bestanden, 21
 - en externe databasebestanden, 19
 - en PACK, 100
 - STRUCTURE EXTENDED, 53-54, 60
 - bewerking van bestanden, 21
- COUNT, 23, **58-59**
- CREATE, **60-63**
 - bestanden aanmaken, 21
 - en APPEND, 42
 - en INSERT, 88
 - en verschillende soorten commando's, 21
 - FROM, 60-62
- DEBUG, 39
- DELETE, **63-64**
 - bestandsbewerking, 22
 - gegevens wijzigen, 21
 - en PACK, 63
 - en RECALL, 63, 106
 - WHILE, 63
- DELETE FILE, 63
- DELETE NEXT, 63
- DELETE RECORD, 63
- DISPLAY, **65-66**
 - en FIELDS, 65
 - en LIST, 93
- FILES, 65
 - weergave van gegevens, 22
 - geheugenvariabelencommando's, 23
- MEMORY, 65
- OFF, 33, 65
- WHILE, 65
- STATUS, 66
- DO, 7, 22-23, **67-69**
- DO CASE, 67-69
- DO WHILE, 67-69
- EDIT, 21, **70-72**
 - apparaatcommando's, 23, 24
- EJECT, 73
 - apparaatcommando, 23
 - en @, 35

- ELSE, 23, 68, 81
 ENDCASE, 68-69
 ENDDO,
 en commandobestanden, 23
 en DO, 67
 en LOOP, 96
 ENDIF, 23, 81
 ENDTEXT, **142**
 ERASE, **74**
 apparatuurbesturing, 23
 en @, 36
 en READ, 103
 FIND, **75-77**
 en SET, 136
 plaatsingscommando, 22
 FORMAT TO SCREEN, SET -,
 39
 form feed, zie papieropvoer
 geheugenvariabelen, 23
 GO, 78
 GOTO, **22, 78**
 HELP, 80
 hoofdletters voor commando's, 29
 IF, 23, 81
 INDEX, **82-85**
 en creëren van een bestand, 21
 en EDIT, 70
 en FIND, 82
 en PACK, 100
 en REINDEX, 108
 en SORT, 138
 en USE, 147
 INPUT, 23, **86-87**
 INSERT, 21, **88-89**
 INTENSITY, 39
 JOIN, 6, 21, **90-92**
 kleine letters voor commando's, 29
 lengte van commando, 29
 overzicht van commando's, Bijlage
 A
 LIST, **93**
 LIST FILES, 93
 LIST MEMORY, 93
 LIST STATUS, 93
 LIST STRUCTURE, 93
 LOCATE, 94-95
 en CONTINUE, 52
 en FIND, 75
 plaatsingscommando, 22
 LOOP, 23, 96
 MODIFY, 8, 21
 MODIFY COMMAND, 23,
 97-98
 MODIFY STRUCTURE, 21,
 97-98
 NOTE, 99
 PACK, **100-101**
 en INDEX, 82
 wijzigen van gegevens, 21
 plaatsingscommando's, 22
 QUIT, 102
 READ, **103-105**
 en @, 35, 39
 gegevens wijzigen, 21
 gegevens weergeven, 22
 RECALL, 21, 106-107
 REINDEX, 108
 RELEASE, 109
 RELEASE ALL, 109
 REMARK, 110
 RENAME, 22, 111
 REPLACE, **112-114**
 en gegevens wijzigen, 21
 REPORT, **115-124**
 creëren van, 21
 en TOTAL, 143
 gegevens weergeven, 22
 RESET, **125**
 RESTORE, 23, **126**
 RETURN, **127**
 en commandobestand, 23
 en DO, 67
 SAVE, 21, 23, 126, **128**
 SAVE TO en RESTORE, 126
 SAY, 35, 36
 SCREEN, 39
 SELECT, 6, 22, **129-130**
 SELECT PRIMARY, 90, 129
 SELECT SECONDARY, 129
 SET-commando's, **131-136**
 ALTERNATE, 131, 135, 136
 BELL, 132
 CARRY, 42, 133
 COLON, 132
 CONFIRM, 133
 CONSOLE, 131

6... INDEX

- DATE, 136
- DEBUG, 133
- DEFAULT, 135
- DELETED, 133
- ECHO, 131
- EJECT, 133
- ESCAPE, 132; zie ook
 - < escape >
- EXACT, 132
- FORMAT, 37, 134
- FORMAT TO, 35, 36, 70, 103, 134
- FORMAT TO PRINT, 35, 37, **134**
- FORMAT TO SCREEN, **134**
 - en @, 35, 36
 - en ERASE, 103
- HEADING, 134
- HEADING TO, 134
- INDEX TO, 136
- INTENSITY, 132
- LINKAGE, 129, 132
- MARGIN TO, 136
- PRINT, 131
- RAW, 33, 133
- SCREEN en EDIT, 70, 132
- SCREEN en ERASE, 74, 103
- STEP, 131
- TALK, 131
- SKIP, 21, 76, **137**
- SORT, 22, **138-139**
- STORE, **140**
 - en REPLACE, 112
 - geheugenvariabelen-commando, 23
- SUM, 22, 23, **141**
- symbooldefinities, 26-27
- taalregels, 28
- TEXT, **142**
- TOTAL, 21, **143-144**
- UPDATE, **145-146**
 - gegevens wijzigen, 21
 - RANDOM, 145
- USE, **147**
 - bewerking bestanden, 22
 - vraagtekencommando, 33
- WAIT, **148**
 - in verschillende soorten
 - commando's, 23
 - wat is?, 22, 33
 - 'werkwoord' in commando - definitie, 28
 - commentaar, 8, 99
 - CONFIRM-parameter, 133
 - CONSOLE-parameter, 131
 - constante, definitie van een, 9
 - CONTINUE en LOCATE, 52, 94
 - CONTINUE, 22, **52**
 - < control >, 3
 - control characters, zie besturingstekens
 - controletoetsen in BROWSE, 48
 - COORD, 70
 - COPY, **53-57**
 - en bestanden creëren, 21
 - en PACK, 100
 - en SDF-bestanden, 20, 22
 - en STRUCTURE EXTENDED, 60
 - overbrengen van databases, 22
 - COPY DELIMITED, **53-57**
 - correcties, 1
 - COUNT, **58-59**
 - weergave van gegevens, 22
 - geheugenvariabelen, 23
 - WHILE, 58
 - CP/M:
 - conventies, 20
 - bestanden, standaard, 5
 - opnieuw instellen, 125
 - commando's, 102
 - en RESET, 125
 - standaard disk-drive, 135
 - CP/M-86, 4
 - CREATE, **60-62**
 - en APPEND, 41
 - en soorten commando's, 22
 - FROM, 60
 - creëren van bestanden, 21
 - creëren van structuren, 60
 - < control > -A in BROWSE, 48
 - < control > -B in BROWSE, 48
 - < control > -C in BROWSE, 48
 - < control > -D in BROWSE, 48
 - < control > -E in BROWSE, 48
 - < control > -F in BROWSE, 48

- < control > -G in BROWSE, 48
 - < control > -H – wissen laatst ingetoetste teken, 3
 - < control > -N, 97
 - < control > -P – printer-schakelaar, 3
 - < control > -Q:
 - in BROWSE, 48
 - in MODIFY, 98
 - < control > -R in BROWSE, 48
 - < control > -S in BROWSE, 48
 - < control > -T in MODIFY, 97
 - < control > -U:
 - wissen van huidige regel, 3
 - in BROWSE, 48
 - < control > -W:
 - in BROWSE, 48
 - in MODIFY, 97
 - < control > -X:
 - wissen van huidige regel, 3
 - in BROWSE, 48
 - < control > -Z in BROWSE, 48
 - cursor:
 - besturing, full-screen
 - controltoetsen, 24–25
- D -**
- dalende volgorde, 138
 - data, geldige, 1
 - databasebestanden, 5
 - DATE(), 13
 - datum-functie, 13
 - dBASE II:
 - bestanden, 5
 - distributiediskette, 1
 - DEBUG-parameter, 39, 133
 - decimale getallen, veldlengte, 60
 - deelreeks:
 - functie, 11
 - operator \$, 17
 - zoekfunctie voor deelreeksen, 13
 - deelteken, 16
 - definities:
 - ALL, 27
 - begrenzer, 26
 - bereik, 27
 - bestand, 5, 26
 - character string, zie tekenreeks
 - commando, 26
 - commandolengte, 26
 - commandowerkwoord, 28
 - constante, 9
 - FOR, 28
 - formatbestand, 27
 - functie, 10
 - gegevensitem, 9
 - geheugenvar, 27
 - gegevensveld, naam, 5
 - gereserveerd woord, 29
 - indexbestand, 27
 - lengte van commando, 27
 - literal, 9
 - lijst, 26
 - lijst geheugenvar, 27
 - lijst uitdr., 26
 - lijst velden, 26
 - logische constante, 9
 - macro's, 19
 - n, 27
 - NEXT n, 28
 - opdracht, 26
 - RECORD n, 28
 - scheidingsteken, 26
 - sleutel, 27
 - symbolen, 26–28
 - tekenreeks, 26
 - tekenreeks-constante, 9
 - tekenreeks en ACCEPT, 40
 - uitdr., 26
 - uitdrukking, 9, 26
 - variabelen, 9
 - veld, 26
 - WHILE, 28
 - DELETE,, 63–64
 - bestandsbewerking, 22
 - wijzigen van gegevens, 21
 - en PACK, 100
 - en RECALL, 106
 - DELETE FILE, 63
 - DELETE NEXT, 63
 - DELETE RECORD, 63
 - DELIMITED:
 - en APPEND, 41
 - sleutelwoord, 41, 53
 - als optie, 20
 - DISPLAY, 65–66

8 ... INDEX

geheugenvariabelen, 22
geselecteerde gegevens, 22
i.t.t. LIST, 93
DISPLAY FILES, 65
DISPLAY MEMORY, 65
DISPLAY OFF, 33
DISPLAY STATUS, 65
DISPLAY STRUCTURE, **65-66**,
129
distributieschijf, 1
DO CASE, 67
DO, 7, 22, 23, **67-69**
DO WHILE, 67-69, 81
DO WHILE en LOOP, 96
dubbele aanhalingstekens als
scheidingstekens, 86
dubbele punt, zie COLON

- E -

ECHO-parameter, 131
EDIT:
commando, 21, **70-72**
cursorbesturing, 24
editen (zie ook wijzigen):
gegevens, 21
eerste/laatste record, 78
einde bestand en RETURN, 127
einde-bestand-functie, 13
EJECT, **73**
en @, 35
apparatuurcommando, 23
EJECT-parameter, 133
ELSE, 23, 81
END CASE, 67-69
ENDDO, 23, 67, 96
ENDIF, 23, 81
END OF FILE, 97, zie ook einde
bestand
END OF LOCATE, 94
END RUN dBASE, 102
ENDTEXT, **142**
enkele aanhalingstekens als
scheidingstekens, 86
EOF, einde-bestand-functie, 13
ERASE, **74**
en @-commando, 35, 74
of CLEAR GETS, 51

scherm wissen, 23
en READ, 106
< escape > - verlaten van
langlopende commando's, 3
ESCAPE-parameter, 132
EXACT-parameter, 132
EXCEPT in RELEASE-commando,
109
EXTENDED, COPY
STRUCTURE -, 53-54, 60

- F -

FIELDS:
en BROWSE, 48
en DISPLAY, 65
en JOIN, 90
en LIST, 93
en TOTAL, 143
FILES, DISPLAY -, **65**
FIND
eigenschappen, 75-77
plaatsingscommando, 22
en SET, 136
FOR-zin:
definitie, 28
en COUNT, 58
en RECALL, 106
en REPLACE, 112
en SUM, 141
form feed, zie papierproer
formatbestand, 27
formulierbestand, 115
FORM-zin, 115
format(s), 38
-tekens, 38
-bestanden, 8
system data format, 41, 53
FORMAT TO SCREEN, 39
FORTRAN, 20
FORTRAN, koppeling aan, 20
foutmeldingen, zie Bijlage C
FROM, APPEND -, 41
full-screen:
en APPEND, 42
en BROWSE, 48
cursorbesturing, 24, 98
en EDIT, 70

en MODIFY COMMAND, 97
 en READ, 103
 functie(s):
 bestand-functie, 15
 DATE(), 14
 deelreeks, 11
 deelreeks zoeken, 13
 definitie, 10
 gewist record, 13
 einde bestand EOF, 13
 FILE, 15
 geheel getal, 10
 getal naar teken, 14
 hoofdletter, 14
 lengte, 12
 lijst van, Bijlage A
 rang, 12
 recordnummer, 10
 reeks naar numeriek, 12
 substring, zie deelreeks
 TRIM, 15, 82
 TYPE, 15

- G -

geldige data, voorbeelden van, 1
 gegevens:
 toevoegen van, 21
 veranderingen in velden, 70
 weergave van, 115
 veldvervangings, 112
 definitie, 9
 format van gegevensrecord, 5
 systeemgegevensformat, 20
 soorten gegevens, 6
 zie ook data
 gegevensveld, naam, 5
 geheugenbestanden, 6
 geheugenvar, 27, 40
 lijst -, definitie, 27
 geheugenvariabele(n):
 in FIND-argument, 75
 opslaan van, 128
 -commando's, 23
 definitie, 6
 RELEASE, 109
 namen van, 29
 wijzigen van, 140

geïndexeerd bestand:
 en EDIT, 71
 en FIND, 75
 geneste commandobestanden, 7
 gereserveerd woord - definitie, 29
 GET, 35, 39, 103
 GETS-sleutelwoord in
 CLEAR-commando, 51
 gewist record-functie, 13
 GO, 22, 78
 GO BOTTOM, 78
 GO TOP, 78
 GOTO, 22, 78
 grammaticale regels, zie regels
 grenswaarden, zie limietwaarden
 groter-dan-teken, 17
 groter-dan-of-gelijk-aan-teken, 17

- H -

haakjes voor prioriteit, 16
 HELP, 80
 hoofdletter-functie, 14
 hoofdletters voor commando's en
 sleutelwoorden, 29
 huidige recordpointer, 78, 137

- I -

IF, 23, 81
 IF..ELSE..ENDIF, 81
 INDEX, **82-85**
 en EDIT, 70
 en LOCATE, 94
 en PACK, 100
 en REINDEX, 108
 en SORT, 138
 en USE, 147
 indexbestand(en), 8
 definitie, 27
 indexsleutel en REPLACE, 112
 indexpointer, 136
 INPUT, 23, **86-87**
 INSERT, 21, 25, **88-89**
 INT - geheel getal-functie, 10
 INTENSITY, 39, 132
 interface, zie koppeling
 is-gelijk-aan-teken, 17

10 ... INDEX

- J -

JOIN, 6, 21, **90-92**

- K -

'karakter', zie teken

kleine letters:

voor commando's, 29

voor sleutelwoorden, 29

kleiner-dan-teken, 17

kleiner-dan-of-gelijk-aan-teken, 17

koppeling aan andere

programmatuur, 20

- L -

laatste/eerste record, 78

lengte-functie, 12

lengte:

van commando, 29

van veld, 60

lijst - definitie, 26, 27

lijst geheugenvar, 27

lijst uitdrukkingen, 26

LIKE in RELEASE-commando, 109

limietwaarden van dBASE II, zie

Bijlage B

LINKAGE-parameter, 132

LIST, **93**

LIST FILES, 93

LIKE in commando LIST FILES,

93

LIST MEMORY, 93

LIST STATUS, 93

LIST STRUCTURE, 93

literals, zie constante

LOAD, 32

LOCATE, **94-95**

en CONTINUE, 52

en FIND, 75

en INDEX, 94

plaatsingscommando's, 22

logische constante - definitie, 9

logische bewerkingen, 6, 9, 15, 17, 60

logische operatoren, prioriteit, 18

LOOP, 23, **96**

lus, DO WHILE-, 96

- M -

maalteken, 16

machinetaal-interface, 31

macro-substitutie, **19**

MEMORY, DISPLAY -, 65, 93

min-teken, 16

MODIFYCOMMAND, 8, 21, 23, **97**

MODIFY, cursorbesturing voor, 25

MODIFY STRUCTURE, 97

MSDOS besturingssysteem, 4

- N -

n - definitie, 27

namen, veld-, 60

nesten van dBASE-opdrachten, 29, 30

NEXT-bepaling, 94

NEXT n - definitie, 28

niet-dBASE processors, 20

niet-gelijk-aan-teken, **17**

NOTE, 99

NOUPDATE-optie, 103, 112

numeriek veld, 6

numeriek veld in CREATE, 60

numeriek-naar-teken-functie, 14

numeriek gegevenstype, 60

numerieke uitdrukkingen, optellen

van, 141

- O -

operating system, zie

besturingssysteem

operatoren, prioriteit van, 16

operatoren, zie ook bewerkingen

opdracht - definitie, 26

opmerkingen, zie commentaar

opslaan van constanten en variabelen,

131

optellen, 16

optellen, zie ook SUM

OTHERWISE, 69

- P -

PACK, 100

en wijzigen van gegevens, 21
 en INDEX, 82
 en DELETE, 63
 papieropvoer, 73
 parameters:

ALTERNATE, 131
 BELL, 132
 CALL, 32
 CARRY, 133
 COLON, 132
 CONFIRM, 133
 CONSOLF, 131
 DATE, 136
 DEBUG, 133
 DEFAULT, 135
 DELETED, 133
 ECHO, 131
 EJECT, 133
 ESCAPE, 132
 EXACT, 132
 FORMAT, 134
 HEADING, 134
 INDEX, 136
 INTENSITY, 132
 LINKAGE, 132
 MARGIN, 136
 PRINT, 131
 RAW, 133
 SCREEN, 132
 STEP, 131
 TALK, 131
 PASCAL, 20
 PEEK, 32
 PICTURE, 38
 plaatsingscommando's, 22
 PLAIN in commando REPORT,
 115, 123, 124
 plusteken, 16
 pointer, record-, 137
 pointer, index-, 136
 POKE, 32
 PRINT-parameter, 131
 prioriteit van rekenkundige, logische
 en reeksoperatoren, 18
 prompt, dBASE-, 1
 punt, als dBASE-prompt, 1

- Q -

QUIT, 102

- R -

RANDOM, 145
 rang-functie, 12
 RANK, 12
 RAW, 133
 READ, **103-105**
 en @, 35, 39
 gegevens wijzigen, 21
 gegevens weergeven, 22
 RECALL, 21, 63, **106-107**
 RECORD n - definitie, 28
 record
 format van gegevens-, 5
 GOTO, 78
 -nummer-functie, 10
 verwijderen van, 100
 structuur-, 5
 reeks:
 -aaneenschakeling, 18
 -functie, 10
 -bewerkingen, 18
 -operatoren, 18
 reeks-naar-numeriek-functie, 11
 regel wissen, 3
 regels, voor commando's, 28-30
 REINDEX, 108
 rekenkundige operatoren, 16
 prioriteit, 18
 RELEASE ALL, 109
 REMARK, 110
 RENAME, 111
 REPLACE, **112-114**
 gegevens wijzigen, 21
 en SELECT, 129
 en STORE, 140
 en UPDATE, 145
 REPORT, **115-124**
 creëren van rapportbestand, 21
 en TOTAL, 143
 weergave van gegevens, 22
 rapportbestanden, 7
 en rapporten maken, 117
 RESET, 125

12 ... INDEX

- RESTORE, 23, **126**, 128
RETURN, **127**
 en commandobestand, 23
 en DO, 67
< rubout >, 3, 48
- S -**
- samenvoegen van databases, 90-92
SAVE, 21, 23, 126, **128**
SAY, 35-36, 38, 39
scheidingsteken - definitie, 26
scheidingsteken, 86
scherm, wissen, 23, 74
schijfcapaciteit en CP/M, 100
schrijven verwisselen, 125
SCREEN, parameter, 39, 132
SDF:
 en APPEND, 41
 en COPY, 53
 optic, 20, 53
SELECT, 6, 22, **129-130**
SELECT PRIMARY, 90, 129
SELECT SECONDARY, 129
SET, **131-136**
 ALTERNATE TO, 136
 CALL, 32
 CARRY, 42, 133
 en commandobestanden, 23, 129, 131
 DATE TO, 14, 117, 136
 DEFAULT TO, 135
 DELETE en FIND, 76
 DELETED, SET en COUNT, 58
 en DISPLAY STATUS, 65
 EJECT, 117
 EXACT, 76
 FORMAT, 37
 FORMAT TO, 35, 36, 70, 103
 en EDIT, 70
 PRINT, 35-37
 SCREEN,
 en @, 35-36
 en READ, 103
 HEADING TO, 118, 134
 LINKAGE, 129
 MARGIN TO, 136
 PRINT, 131
- RAW, 33
SCREEN, 74, 132
 en STATUS, 65
SKIP, 22, 76, 137
sleutel - definitie, 27
sleutelwoord(en):
 afkortingen voor, 29
 DELIMITED, 41, 53
 GETS, 51
 hoofdletters voor, 29
 kleine letters voor, 29
sluiten van databasebestanden, 102
soorten commando's, 21-23
SORT, 22, **138**
sorteervolgorde, ASCII, 138
spatie-terugtoets - wissen van laatste teken, 3
standaard drive, CP/M, 135
standaardwaarden in formatbestand, 115
standaard CP/M-bestanden, 5
STATUS, DISPLAY -, 65, 93
STEP-parameter, 131
stijgende sortering, 138
STORE, **140**
 geheugenvariabele-commando, 23
 en REPLACE, 112
STR, 10
STRUCTURE:
 -bepaling, 53
 DISPLAY -, 66
 LIST -, 93
 MODIFY -, 97-98
structuur (-turen):
 creëren van, 60
 van record, 5
substring zie deelreeks
subtotaliseren, 119, 143
SUM, 23, **141**
symbooldefinities, 26
SYNTAX ERROR - foutmelding, 1
systeem:
 -gegevensformat, SDF, 20
 -vereisten, 4
systeem-gegevensformat:
 en APPEND, 41
 SDF, 53
 en COPY, 53

- T -

TALK-parameter, 131
 tekenreeks:
 en ACCEPT, 40
 -constante, 9
 veld, 5
 veld in CREATE, 60
 teken wissen, 3
 tekstuitvoerbestand, 8
 TEXT, **142**
 TO, 102, 148
 toevoegen van gegevens, 21
 TOP, GOTO, 78
 TOTAL, 21, **143-144**
 TRIM-functie, **15**, 82
 TYPE - gegevenstype-functie, 15

- U -

uitdrukking - definitie, 26
 uitdrukking(en):
 definitie, 9
 lijst -, 26
 uitvoering, voorwaardelijke, 81
 uitvoeringsvolgorde van operatoren,
 18
 UPDATE, **145-146**
 gegevens wijzigen, 21
 USE, **147**

- V -

variabelen:
 definitie, 9
 GET, 39
 geheugen-, 27, 40
 veld(en):
 definitie, 26
 lengte, 60
 lijst, definitie, 26
 -grootte, 6
 veldlengte, 60
 veldnamen, 60
 tekens voor, 29
 venster, 25, 48
 verbeteringen - voorbeelden, 2

vereisten, systeem-, 4
 vergelijking, 17
 verschuiven van venster, 25, 48
 vermenigvuldiging, 16
 verwisselen van schijven, 125
 voorwaardelijke uitvoering, 81
 vraagteken-commando, 22, 33

- W -

WAIT, 148
 in verschillende soorten
 commando's, 23
 wat is ...?-commando, 22, 33
 werkwoord in commando, 28
 WHILE:
 en COUNT, 58
 definitie, 28
 DO, 67
 RECALL, 106
 en REPLACE, 112
 wijzigen van bestandsnaam, 111
 wijzigen van gegevens, 21
 wijzigingen in gegevensvelden, 70
 wismarkering, verwijderen van,
 106
 wissen van huidige regel, 3
 wissen van scherm, 23
 woordscheiding, 28

- X -

X als formatteken, 38

- Z -

Z-80 microprocessor, 4
 zinsdelen:
 ADDITIVE, 126
 BEFORE, 88
 BLANK, 41, 88
 FIELDS, 90, 143
 FOR, 28
 FORM, 115
 GET, 36, 103
 SAY, 35-36
 TO, 102
 WHILE, 28



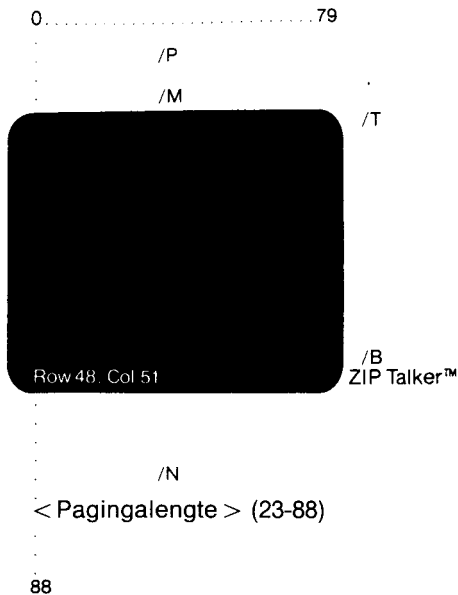
ZIP

ASHTON · TATE ■TM

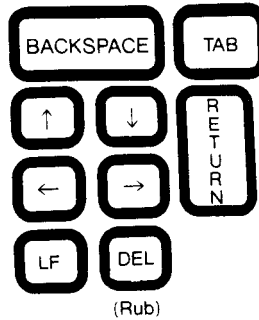
© 1985 Ashton-Tate
Reproduction by any method is strictly prohibited

ZIP SAMENGEVAT:

ZIP maakt van uw beeldscherm een "venster" voor het aanmaken van 23-regels lange invoerschermen en tot 88 regels lange printouts:



Uw ASCII toetsenbord werkt als een toetsenbord, deze toetsen doen precies wat u van ze verwacht:



Typ het aantal tekens (max. 50) voor SCHERM WISSEN & HOMEPOSITIE CURSOR, CURSORPOSITIONERING en CURSORBESTURINGSTOETSEN

ZIP's SCHERMCOMMANDO'S (Direct in te toetsen en te veranderen)

/	: Commandosymbool	/Q	: Keert terug naar besturingssysteem
/C	: Centreert tekst op de regel	//	: Hulpscherm
/T	: (Top) bovenste schermregel	/< < tab >	: Rand van het scherm
/I	: Invoegmodus	/B	: Onderste schermregel
/M	: Midden van regel/scherm	/D	: Wist een teken
/A	: Rij of kolom invoegen	/K	: Wist een rij of kolom
/H	: Horizontale lijn tekenen/wissen	/V	: Verticale lijn tekenen/wissen
/N	: Volgend scherm	/P	: Vorig scherm
/F	: Eerste scherm	/L	: Laatste scherm
/E	: Wist het werkbestand	/S	: Slaat het bestand op, schrijft programma

Voor dBASE II:

@	: SAY variabele	#	: GET variabele
[]	: VOORBEHOUDEN voor GENESTE dBASE II COMMANDO'S		

Dynamische waarden die u tijdens een ZIP sessie kunt veranderen:

.	: < V > erticale markering	-	: < H > orizontale markering
5	: < T > ab-afstand (1-79)	88:	< P > aginalengte (23-88)
		40:	< M > arge (0-132)

dBASE II werkt nu met ZIP < tm > .

dBASE II was altijd al het krachtigste en meest gebruikersvriendelijke systeem voor relationeel databasemanagement op micro's. Met de nieuwe ZIP werkt dBASE II echter nog sneller dan ooit.

Met ZIP wordt de opbouw van schermen en outputformulieren in uw programma's veel eenvoudiger, een pluspunt dat u zeker op prijs zult stellen. De voornaamste kenmerken van ZIP zijn:

- U kunt de *outputformulieren voor uw printer* en de *inputformats voor uw scherm* direct op uw beeldscherm maken omdat ZIP tot 88 regels kan verwerken.
- *ZIP creëert automatisch de programmacode:* ZIP schrijft een READ aan het eind van elk bestand waarin GET's worden gebruikt en een READ voor elke 64 GET-instructies in langere bestanden, valideert variabelenamen, schrijft .FMT bestanden of schrijft volledige .CMD bestanden met ERASE of SET FORMAT TO PRINT/SCREEN, SET MARGIN TO <n> en RETURN, zonder dat u iets hoeft te doen.
- *Neem dBASE II commando's in uw formatbestanden op:* met ZIP kunt u formats en bewerkingen samen wijzigen en opslaan.
- *Gebruik ieder willekeurig ASCII-bestand:* als u formulieren heeft die met een tekstverwerker zijn voorbereid, dan kunt u deze bestanden (max. 88 regels) laden en ZIP de code laten schrijven.
- *De ZIP Talker™*, een regel onderaan het scherm, geeft aan op welke rij en kolom uw cursor zich bevindt, zodat u de formatposities snel en makkelijk kunt instellen. Op deze regel verschijnen eveneens de prompts tijdens de ZIP-sessie.
- Met ZIP kunt u de symbolen voor horizontale en verticale lijnen, de tab-afstand, de paginalengte en de printermarge *tijdens de opbouw* van een formulier of een schermformat soepel wijzigen.
- Met de symbolen Horizontaal en Verticaal kunt u *lijnen trekken en wissen*, zodat u complete kaders, rijen of kolommen snel kunt tekenen of wissen.
- Met ZIP werkt u veel sneller omdat u *geen control-toetsen* hoeft in te drukken. De werking van uw toetsenbord blijft ongewijzigd: < Return >, < Line Feed >, < Delete >, < Backspace >, < Tab > en de cursortoetsen doen precies wat u van ze verwacht.
- U kunt *de standaardwaarden zelf wijzigen* en deze precies zo instellen als u het wenst.

INHOUD

ZIP installeren	3
U heeft een andere terminal	4
Op de klant afgestemde installatie	5
Wijziging van de afkortingen en symbolen	7
Wat ZIP doet, kort samengevat	10
Werken met ZIP	11
Symbolen voor horizontale/verticale lijnen	12
Hulpinformatie, opslag van uw bestanden	15
Aanmaak van een nieuw bestand, Quit	17
Nooit meer @..SAY..GET	18
Opgenomen dBASE II commando's	19
Praktische tip: kleine velden, lange namen	20
Andere bestandstypes, opmaak	22
dBASE II functies, dynamische variabelen	23
Andere ZIP commando's en symbolen	24

ZIP INSTALLEREN

Op de distributieschijf met enkele schrijfdichtheid staan ZIPIN.COM, ZIP.COM, ZSCRN.OVL EN DGEN.OVL.

Maak op een werkschijf en een reserve-schijf een kopie van de distributieschijf en berg de distributieschijf *alstublieft* op een veilige plaats op.

Daarna steekt u een schijf waarop ZIP.COM staat in de drive. Het installatieprogramma ZIPIN.COM mag in iedere willekeurige drive zitten. Typ:

< drive: > ZIPIN

Als uw terminal is opgenomen in de lijst die ZIPIN op het eerste scherm toont, dan typt u het getal dat naast uw terminal vermeld is, gevolgd door < Return > . Als dit niet het geval is, kiest u 0 (nul) en volgt u de installatieprocedure die hieronder begint.

APPLE II en III: als u de Apple-versie heeft, dan kunt u het 24 x 40 scherm, het 24 x 80 scherm of de Apple III via het menu installeren.

Osborne 1: De Osborne-versie is van tevoren geïnstalleerd, zodat zoveel mogelijk schijfruimte voor u vrij blijft.

Druk nog eens op < return > om uw keuze te bevestigen.

Het volgende scherm toont de ingestelde waarden die u geselecteerd heeft. Op dit punt kunt u alle waarden nog wijzigen.

Bij de Hazeltine 1500 bijvoorbeeld wordt de cursor met de controltoets bestuurd aangezien deze computer geen aparte cursortoetsen heeft. Deze besturing is overigens dezelfde als voor dBASE II:

UP	= <control >-E
DOWN	= <control >-X
LEFT	= <control >-S
RIGHT	= <control >-D

Als u enkele andere combinaties (inclusief meervoudige tekenreeksen) wilt gebruiken, kiest u het getal naast de te veranderen ingestelde waarde en voert u de nieuwe gegevens in zoals wordt beschreven in de volgende paragrafen over de op de klant afgestemde installatie.

OPMERKINGEN BIJ DE INSTALLATIE:

1. De optionele Automatische Scrolling (auto scroll) functie moet INgeschakeld zijn (bv. TRS-80 II met P & T CP/M).
2. De optionele Lokale Echo-functie (local echo) moet UITgeschakeld zijn (bv. HP 125).

4 ... ZIP

Als uw terminal niet in de lijst vermeld is, zoekt u de bij uw terminal horende code-sequenties op voor:

- SCHERM WISSEN (CLEAR SCREEN) EN HOME-POSITIE CURSOR
- DIREKTE CURSORBESTURING (en de BASISWAARDE) (BIAS/OFFSET)
- PIJLEN (CURSORTOETSEN)
- SET UP/RESET (gewoonlijk niet vereist)

Alles draait hier om *sequenties*. De meeste terminals hebben enkele afzonderlijke toetsen die meervoudige tekens zenden wanneer ze worden ingedrukt. Zo zal "CLEAR" waarschijnlijk minstens twee tekens zenden: een teken om het scherm schoon te maken en een ander om de cursor naar de beginpositie te brengen. Het is mogelijk dat de pijltoetsen eveneens meer dan één teken zenden.

Zulke toetsen moeten als een sequentie worden ingevoerd en kunnen niet in gebruik genomen worden door op de gewenste toets te drukken.

Zoekt u die even op in de handleiding van uw terminal en typ de *afzonderlijke tekens* uit de sequenties wanneer ZIPIN hierom vraagt. De besturingssequenties mogen elk net zo lang zijn als nodig, mits de totale lengte van alle sequenties niet meer dan 50 tekens bedraagt.

Alle andere *speciale toetsen* en sequenties die uniek zijn voor uw terminal kunnen al dan niet functioneren; voor een efficiënt gebruik van ZIP heeft u ze in ieder geval niet nodig.

Tijdens de installatie aanvaardt ZIPIN alleen de *tekens* in een sequentie en *niet* de codes die deze tekens weergeven. Als uw codes als decimaal of hexadecimaal getal of als naam gegeven zijn, dan gebruikt u voor de vertaling ervan de volgende tabel:

ALS DE CODE IS:		TYPT U:		ALS DE CODE IS:		TYPT U:	
DECIMAAL	HEXA-DECIMAAL	NAAM	TOETS	DECIMAAL	HEXA-DECIMAAL	NAAM	TOETS
0	0	NUL	ctl-@	16	10	DLE	ctl-P
1	1	SOH	ctl-A	17	11	DC1	ctl-Q
2	2	STX	ctl-B	18	12	DC2	ctl-R
3	3	ETX	ctl-C	19	13	DC3	ctl-S
4	4	EOT	ctl-D	20	14	DC4	ctl-T
5	5	ENQ	ctl-E	21	15	NAK	ctl-U
6	6	ACK	ctl-F	22	16	SYN	ctl-V
7	7	BEL	ctl-G	23	17	ETB	ctl-W
8	8	BS	ctl-H/BS	24	18	CAN	ctl-X
9	9	HT	ctl-I/Tab	25	19	EM	ctl-Y
10	A	LF	ctl-J/LF	26	1A	SUB	ctl-Z
11	B	VT	ctl-K	27	1B	ESC	ctl-E
12	C	FF	ctl-L	28	1C	FS	ctl-\
13	D	CR	< return >	29	1D	GS	ctl-ESC
14	E	SO	ctl-N	30	1E	RS	ctl-^
15	F	SI	ctl-O	31	1F	US	ctl- _

Bijvoorbeeld: om het ESCAPE-teken in te voeren drukt u ofwel op de <escape> toets ofwel op <control>-[. Om een besturingsteken in te voeren houdt u de CONTROL-toets (soms "ALT" genaamd) *ingedrukt* terwijl u op de lettertoets of de grafische toets drukt.

<return> *mag nooit gebruikt worden in besturings-sequenties.*

Om met de op de klant afgestemde installatie van start te gaan, typt u 0 (nul) als terminalselectie en drukt u op <return> om uw keuze te bevestigen.

Nu toont het scherm de waarden die voor de terminalbesturings-sequenties werden ingesteld. De lengte van de afzonderlijke sequenties is willekeurig, voor zover het totaal van alle sequenties niet meer dan 50 tekens bedraagt.

Om te beginnen kiest u 1, de sequentie om het SCHERM SCHOON TE MAKEN EN DE CURSOR NAAR DE BEGINPOSITIE TE BRENGEN, waarna u op <return> drukt.

ZIP toont de ingestelde waarde en vraagt of u deze wilt veranderen. Druk op C.

U kunt nu zoveel tekens invullen als u wilt. Als u klaar bent, drukt u op <return> om de invoer te bevestigen.

Daarna kiest u 2 voor DIREKTE CURSORBESTURING.

Bij de invoer van deze sequentie typt u 0 (nul) op de tekenposities waar de rij- en kolomgegevens moeten komen. Vervolens vraagt ZIP u naar de rij en de kolom.

De meeste terminals hebben een numerieke BASIS die bij de rij- en kolomwaarden wordt opgeteld. Deze basis is vaak gelijk aan 32, maar kan ook 0, 128 of een andere waarde zijn.

Voor de cursorpositionering gebruikt men meestal de ASCII-techniek (de posities worden als volledige getallen gezonden) of de ANSI-methode (de positie-*cijfers* worden afzonderlijk gezonden). Bij twijfel gebruikt u de ANSI-methode. Als deze methode geen resultaat oplevert, kunt u achteraf met ZIPIN de ASCII-methode proberen.

6... ZIP

Sommige terminals hebben een vertraging nodig na het zenden van het Clear- en het Home-signaal, anders is het mogelijk dat het eerste blok van twaalf tekens bovenaan een nieuw scherm niet wordt afgebeeld. De meeste terminals kunnen het ZONDER vertraging stellen (noodzakelijk voor de Hazeltine 1500) en ZIP zal sneller werken wanneer de vertraging niet ingesteld is. Als u niet helemaal zeker bent, installeert u ZIP eerst zonder vertraging. Als de vertraging wél nodig is, kunt u ZIP achteraf altijd nog eens installeren.

Als uw toetsenbord geen cursorbesturingstoetsen heeft, wilt u wellicht de door dBASE II gebruikte codes invoeren (<control>-E voor naar boven, <control>-X voor naar beneden, <control>-S voor naar links en <control>-D voor naar rechts).

Daarna geeft u antwoord op de vraag of uw terminal een *terugspatieertoets* (backspace) heeft *die het teken links van de cursor wist*. Sommige terminals gebruiken het Backspace-signaal als cursor-naar-links toets; hierbij wordt de cursor verplaatst zonder het teken te wissen. Als dit voor uw terminal geldt, typt u "N" zodat uw cursor-naar-links toets net zo blijft functioneren als voorheen. Als u "Y" antwoord, zal deze toets bij het terugspatiëren het teken links van de cursor wissen.

U kunt nu de hoogte en de breedte van uw terminalschermbepalen. ZIP loopt op terminals met schermen van 24 tot 60 rijen hoog en van 40 tot 80 kolommen breed.

WIJZIGING VAN AFKORTINGEN EN SYMBOLEN.

Nadat u de cursortoetsen heeft ingesteld, zal ZIPIN het scherm wissen en de dynamische variabelen afbeelden.

Welke standaardwaarde u ook selecteert, u kunt de Tab-afstand, de Horizontaal- en Verticaal-symbolen, de Paginalengte en de Printermarges op ieder moment veranderen wanneer u met ZIP schermen en formulieren gaat opmaken.

Om de wijzigingen tijdens de installatie aan te brengen, typt u de volgende toetsenreeksen:

C	T	39	om de Tab-afstand te wijzigen
C	V		om het Verticaal-symbool te wijzigen
C	H	*	om het Horizontaal-symbool te wijzigen
C	M	40	om de printer-Marge te wijzigen.

Bent u hiermee klaar, dan typt u F, waarna ZIP alle standaard-commando symbolen en afkortingen afbeeldt.

Als u een commando-afkorting of -symbool wilt veranderen, typt u "C" en drukt u op de symbooltoets die u wilt veranderen, gevolgd door de toets voor het symbool dat ervoor in de plaats komt.

Om bijvoorbeeld het commandosymbool te veranderen:

- Typ C
- Typ /
- Typ \ (als dat het symbool van uw keuze is)

Opmerking: De vierkante haakjes ([]) worden gebruikt als scheidingsteken voor reeksen in de dBASE II-commandobestanden die ZIP voor u schrijft. Dat is de reden dat u geen vierkante haakjes als nieuw symbool mag invoeren.

Zo kunt u het commandosymbool "/" vervangen door "\" of een ander teken (in kleine letters: typt sneller) dat u normaliter nooit gebruikt in de schermen of de printformulieren die u aanmaakt.

Opmerking: Kies nooit het zelfde teken voor meerdere commando's of symbolen. ZIP raakt daardoor totaal in de war. Als u dat per ongeluk toch gedaan heeft, kunt u dat weer goedmaken door de installatieprocedure te herhalen.

**ZODRA DE INSTALLATIE VOLTOOID IS,
HEEFT U ZIPIN NIET LANGER NODIG
OP UW WERKSCHIJF.**

**** Bestand B: Voorbid ****

```
+-----+  
| VOORBEELD |  
+-----+
```

VOORBEELDFAKTUUR

FACTUUR @faktnr

DATUM # datum

KLANT @klant

[USE B:KOSTEN] # [STORE 0 TO TOTAAL]
[DO WHILE .NOT. EOF]@[? JOB:Nr, Omschr, Belastbaar] # [STORE
TOTAAL + BELASTBAAR TO TOTAAL]
[ENDDO]

TOTAAL FACTUUR @Totaal 37,79

```

*VOORBLD.CMD
SET FORMAT TO PRINT
SET MARGIN TO 40
@ 0,23 SAY "+ ----- +"
@ 1,23 SAY "| VOORBEELD |"
@ 2,23 SAY "+ ----- +"
@ 7, 5 SAY "*****"
@ 8, 7 SAY "VOORBEELDFAKTUUR"
@ 9, 5 SAY "*****"
@ 12, 5 SAY "FAKTUUR"
@ 12,13 SAY faktr
@ 12,40 SAY "DATUM"
SET FORMAT TO SCREEN
@ 23,79 SAY " "
@ 23, 0 SAY "datum" GET datum
READ
SET FORMAT TO PRINT
@ 15, 6 SAY "KLANT"
@ 15,13 SAY klant
@ 17, 0 SAY "-----"
@ 17,50 SAY "-----"
USE B: KOSTEN
STORE 0 TO Totaal
DO WHILE .NOT. EOF
? Job:Nr,Omschr,Belastbaar
STORE TOTAAL + BELASTBAAR TO TOTAAL
ENDDO
@ 35, 0 SAY "-----"
@ 35,50 SAY "-----"
@ 37,42 SAY "TOTAAL FAKTUUR"
@ 37,58 SAY Totaal
@ 37,75 SAY "37,79"
SET FORMAT TO SCREEN
@ 23,79 SAY " "
RETURN

```

WAT ZIP DOET, KORT SAMENGEVAT

Het *commandobestand* (*uitbreiding* .CMD) op de vorige pagina werd door ZIP geschreven uitgaande van het voorgaande inputscherm (of -schermen).

Het kader werd opgebouwd met de bevelen die de horizontale en verticale symbolen tekenen en wissen, en automatisch een “+” schrijven op de snijpunten. Beide symbolen kunnen gewijzigd worden tijdens de aanmaak van een formulier of printout.

De tekst, variabelenamen (voorafgegaan door @ of #) en ingesloten commando's (tussen vierkante haakjes) zijn zonder gebruik van besturingsfuncties ingetoetst. Zodra het formulier er naar wens uitzag, heeft de operator “/S” ingetoetst waarna ZIP aan het werk ging.

ZIP heeft het format op PRINT ingesteld, omdat de operator “Yes” geantwoord heeft op de vraag of het formulier moest worden afgedrukt. De printermarge is ingesteld op de door de operator geselecteerde waarde.

Daarna heeft ZIP het formulier doorgenomen en automatisch alle @ < rij,kolom > SAY < tekst > commando's ingeschreven.

Telkens als ZIP een DISPLAY symbool (@) tegenkwam, heeft het @ < rij,kolom > SAY < variabele > geschreven.

ZIP heeft bovendien een GET-commando met bijbehorende code geschreven om de cursor naar de onderste schermregel te brengen en om de naam van de variabele af te beelden tijdens de verwerking van het .CMD bestand.

In dit voorbeeld heeft de operator ook een aantal dBASE II commando's tussen vierkante haakjes opgenomen, waardoor het bestand Voorbld.CMD een combinatie is van vorm en functie.

Daarna heeft ZIP orde op zaken gebracht door SET FORMAT TO SCREEN en RETURN te schrijven, waardoor de cursor in de hoek links onderaan komt te staan zodra het format klaar is. Dit bestand kan nu in elk dBASE II commandobestand gebruikt worden door gewoon: DO VOORBLD aan het andere bestand toe te voegen.

De operator kon het bestand ook opslaan als een *formatbestand* (*uitbreiding* .FMT) door het gewenste type op te geven.

Voor een formatbestand zou ZIP alleen de @ ..SAY statements en de gegevens tussen vierkante haakjes hebben ingeschreven. Het zou de SET statements achterwege gelaten hebben, en de eerste commentaarregel zou door * VOORBLD.FMT vervangen zijn.

Onthoud evenwel dat *formatbestanden alleen commentaar tussen vierkante haakjes mogen bevatten*. ZIP gaat niet na of de ingesloten statements correct zijn.

Op de volgende pagina's laten we zien hoe makkelijk dit alles gaat.

WERKEN MET ZIP

Typ:

ZIP

ZIP toont allereerst het Hulpscherf dat u herinnert aan de commando's en standaardwaarden.

Voorlopig hoeft u de commando's niet uit het hoofd te kennen. Terwijl u aan het werk bent met ZIP, kunt u ze nog eens bekijken door twee keer op de Commandotoets te drukken (// of ieder ander symbool dat u ter vervanging heeft geïnstalleerd). Druk op een willekeurige toets om verder te werken.

Het scherm moet leeg zijn en de ZIP Talker (op de onderste regel van het scherm) moet het volgende melden:

< NEW > or < OLD > file (Q to Quit)?

ZIP gaat pas aan de slag wanneer u geantwoord heeft dat u een nieuw bestand wilt beginnen (door N te typen) of dat u een vroeger bestand wilt bijwerken (door de letter O te typen). Mocht u zich bedenken, dan kunt u terugkeren naar het besturingssysteem door Q in te toetsen. We hebben tot nu toe nog geen bestanden geschreven, en typen dus:

N

De ZIP Talker vraagt naar de naam van het nieuwe bestand:

FILE NAME (drive optional):

Als u geen schijfteenheid opgeeft (letter gevolgd door een dubbele punt), worden de bestanden opgeslagen en uitgelezen op uw huidige drive.

Bestandsnamen mogen tot 14 tekens lang zijn: 2 tekens voor de drive gevolgd door een dubbele punt, 8 tekens voor de bestandsnaam en 4 tekens voor één punt gevolgd door een bestandstype. De naam mag alleen de volgende tekens bevatten:

A-Z a-z 0123456789 \$@ # / en de dubbele punt (:)

ZIP zal een schermbeeld van uw werkbestand opslaan als < naam > .ZIP. Het zal eveneens commandobestanden schrijven als < naam > .CMD en formatbestanden als < naam > .FMT.

Alfabetische tekens mogen zowel in hoofdletters als in kleine letters worden ingevoerd – ZIP zet alle bestandsnamen automatisch om in hoofdletters.

12 ... ZIP

In CP/M moet als tweede teken van de bestandsnaam een dubbele punt gebruikt worden. Als u elders een dubbele punt heeft gebruikt, zal ZIP de bestandsnaam weigeren en u vragen hem opnieuw in te voeren.

Als u na invoer van de bestandsnaam op <return> drukt, verspringt de cursor naar positie 0,0 en meldt de ZIP Talker waar hij zich precies bevindt.

Probeer de tab- of de return-toets even. Als uw toetsenbord een automatische repeteerfunctie heeft, houdt u de toets ingedrukt en ziet u hoe uw cursor regel na regel verspringt tot hij helemaal onderaan op het scherm staat. De Talker volgt zijn positie op de voet.

Nu houdt u de cursor-naar-boven-toets ingedrukt en de cursor begint naar boven te "stappen".

U kunt hem ook veel sneller naar de bovenste schermregel brengen. Typ de volgende drie commando's (in hoofdletters of kleine letters):

/T
/B
/M

Met deze drie toetsen kunt u de cursor snel over het scherm verplaatsen. Als u /T (bovenste schermregel) en /B (onderste schermregel) gebruikt, blijft de cursor in dezelfde kolom, zodat u de ingevoerde gegevens makkelijk kunt schikken.

Het "Midden"-commando brengt de cursor naar het midden van de rij, wat het centreren van titels vereenvoudigt. Als de cursor al in het midden van de rij staat, brengt u hem met /M in het midden van het scherm.

Nu typt u:

/<tab> (druk op de tab-toets)
 /<tab>

De eerste keer verspringt de cursor naar de rechterrand van het scherm, en daarna heen en weer over het scherm.

Zie hoe de ZIP Talker alles in het oog houdt. Toen u het commandosymbool intoetste, liet hij u d.m.v. een symbool op de onderste schermregel weten dat ZIP op een instructie wacht.

Telkens als u met ZIP begint wordt de Tab-afstand automatisch ingesteld op de waarde die u geïnstalleerd heeft (oorspronkelijk 5). De Paginalengte wordt ingesteld op de afmetingen van uw scherm. U kunt de Tab-afstand en de Paginalengte naar believen wijzigen. Hiertoe gebruikt u de volgende drie commando's (wacht tot het Hulpscherf verschenen is na het eerste commando):

```
//
T           (hier wordt het commandosymbool niet gebruikt)
19         (willekeurige waarde tussen 1 en de scherm breedte)
```

Druk op iedere willekeurige toets om terug te keren naar uw werkscherf en druk daarna op de Tab-toets. De ZIP Talker geeft aan dat nu om de 19 posities een tabulatiestop is ingesteld. U kunt deze tabulatie-afstand zo vaak u wilt veranderen door terug te keren naar het Hulpscherf.

De Paginalengte wijzigt u op dezelfde manier als de Tab-afstand. Via het hulpscherf kunt u een willekeurige Paginalengte selecteren vanaf de lengte van één scherm (minimumwaarde) tot 88 regels. Zo kunt u officiële documenten op uw beeldscherf voorbereiden en ze laten afdrucken met een regeldichtheid tot 8 regels per inch op standaard-papier.

Opmerking: Als u tijdens een ZIP-sessie de Paginalengte verkleint, dan worden alle gegevens gewist die de nieuwe paginalengte overschrijden. Deze gegevens zijn dan voorgoed verloren!

Breng de cursor nu naar de linker bovenhoek van het scherm (met < return >, /T) en typ:

```
/H
/V
< tab > twee keer
/V
< tab > twee keer
/V
/H
```

Deze commando's kunnen horizontale en verticale lijnen trekken en wissen en op het snijpunt van de lijnen een "+" (plusteken) plaatsen en wissen. U kunt uw schermen en formulieren nu veel makkelijker "tekenen".

14 ... ZIP

Als de cursor op een Horizontaal-symbool staat wanneer u het Horizontaal-commando invoert, zal ZIP het lijngedeelte rechts van de cursor wissen.

Als de cursor op een "+" staat, kan ZIP op verschillende manieren reageren:

Als de cursor op een "+" staat en er andere tekens links van de cursor staan, wordt het plusteken behouden en wordt het horizontale lijngedeelte rechts van de cursor gewist. Als er geen tekens onmiddellijk links van de cursor staan en het volgende symbool rechts van de cursor geen "+" is, zal ZIP de horizontale lijn wissen en "+" vervangen door de huidige waarde voor het Verticaal-symbool.

Als het symbool tussen andere plustekens staat, blijven *a/* die tekens behouden en verdwijnt alleen het deel van de horizontale lijn dat rechts van het uiterst rechtse "+" stond.

Het Verticaal-commando werkt volgens hetzelfde principe.

Het gebruik van beide commando's is veel eenvoudiger dan u wellicht zult denken, aangezien ZIP alle beslissingen voor u neemt.

Opmerking: Door twee keer na elkaar een Horizontaal- of Verticaal-commando in te voeren (/H /H of /V /V) kunt u snel een volledige rij of kolom wissen.

Nu typt u de volgende commando's (wacht tot het hulpscherf is verschenen):

```
//
P
40
*
```

(huidige waarde voor het Verticaal-symbool)

U heeft een paginalengte van 40 regels gekozen en wilt een sterretje gebruiken om verticale lijnen te trekken. Druk een willekeurige toets in om terug te keren naar het werkscherf.

Met de returntoets brengt u de cursor een paar regels omlaag. Typ /H en daarna /V. In plaats van de verticale lijn te wissen, heeft u met het laatste commando alle symbolen door sterretjes vervangen. De cursor stond op dat ogenblik immers niet op het *huidige* verticaal-symbool.

Nu typt u:

/N (wacht op het nieuwe scherm)
/B

Hierdoor komt u onderaan in het volgende scherm terecht; de ZIP Talker meldt: "Row 39, Col 0" (we hebben een Paginalengte van 40 regels die van 0 tot 39 genummerd zijn). Merk op, dat de laatst getekende verticale lijn helemaal tot onderaan de pagina doorloopt, ook al waren niet alle regels zichtbaar op het moment dat u het commando gaf.

Voor langere formulieren is het beter als u het volgende typt:

/F om naar het EERSTE SCHERM van uw formulier over te gaan,
met rij 0 boven in het scherm; of
/L om naar het LAATSTE SCHERM van uw formulier over te gaan,
met de laatste rij onder in het scherm.

Om de ingevoerde gegevens op te slaan typt u;

/S

De ZIP Talker antwoordt met volgende boodschap:

SAVE < name > as CMD or as FMT file (C, F or stop)?

Als u een andere toets dan "C" of "F" indrukt, dan zal ZIP u weer naar het werkbestand overbrengen.

Als u "C" typt, vraag ZIP:

Is this form to be printed (Y or N)?

ZIP beschouwt alle toetsen anders dan "Y" als een 'Neen'.

Daarna geeft ZIP u de gelegenheid om de bestandsnaam te veranderen (zowel voor .CMD- als .FMT-bestanden):

File < name > : do you want to change its name (Y or N)?

Als u "Y" antwoordt, vraagt ZIP naar de nieuwe naam. Om reserve-kopieën te maken op een andere drive, kunt u ook een andere schijf eenheid opgeven als onderdeel van de nieuwe naam. Of u drukt een andere toets in, waarna ZIP het bestand onder zijn huidige naam opslaat.

ZIP slaat eerst een schermbeeld op van het scherm of printformat dat u heeft opgebouwd. De Talker meldt dit met de volgende boodschap:

Writing screen image < name > .ZIP.

16 ... ZIP

Nadat het schermbeeld opgeslagen is, schrijft ZIP nog een afdrukbare kopie van dat schermbeeld. De Talker meldt dit met:

Writing printable file < name > .ZPR.

Als ZIP een dBASE II commandobestand schrijft, voegt het eerst de bestandsnaam in als commentaar (* < NAAM > .CMD of * < NAAM > .FMT).

Als het bestand een .CMD bestand is, dan schrijft ZIP

ERASE (als het om een schermformaat gaat)

of

SET FORMAT TO PRINT
SET MARGIN TO xx (als het om een printformulier gaat)

Voor beide bestandstypes (.CMD en .FMT) schrijft ZIP alle @..SAY en GET statements en voegt het om de 64 GET's een READ in in uw bestand. De ZIP Talker geeft aan op welke rij(en) dit is gebeurd.

ZIP plaatst een READ onderaan de formatbestanden, of SET FORMAT TO SCREEN voor printbestanden vlak vóór de laatste RETURN.

Het oorspronkelijke scherm verandert niet wanneer ZIP uw werkbestand opgeslagen heeft en het programma geschreven heeft: u kunt het verder wijzigen en onder een andere bestandsnaam bewaren.

Wanneer ZIP een GET in een printformulier tegenkomt, vraagt de Talker:

"GET" in PRINTOUT: Okay (Y or N)?

en wacht hij op uw antwoord. Als u de GET opzettelijk heeft gebruikt, typt u "Y" en ZIP zal het bestand < naam > .CMD verder schrijven.

Als de GET per ongeluk in uw bestand gesloten is, drukt u op een willekeurige andere toets. ZIP beëindigt het .CMD bestand, geeft het de markering INCOMPLETE COMMAND FILE en brengt uw cursor naar het foutieve GET-symbool zodat u het kunt veranderen of weghalen.

Opmerking: U moet uw programma opnieuw opslaan (/S) omdat het voortijdig onderbroken werd. Dit neemt enkele seconden in beslag, maar ZIP bespaart u in ruil daarvoor veel tijd bij het maken van uw applicaties.

ZIP valideert uw variabelenamen om ervoor te zorgen dat de geschreven commandobestanden in uw dBASE II programma's verwerkt kunnen worden.

Wanneer u een variabelenaam laat beginnen met een teken dat door dBASE II niet wordt geaccepteerd (alles buiten het alfabet, cijfers en de dubbele punt), verschijnt de volgende boodschap:

NO VARIABLE: continue (Y or N)?

Door op een andere toets dan de "Y" (hoofdletter of kleine letter) te drukken, wordt de Save beëindigd en komt uw cursor op de fout in uw werkbestand te staan, zodat u deze snel kunt verhelpen.

De ZIP Talker waarschuwt u ook wanneer een variabelenaam met een dubbele punt begint of eindigt:

HANGING COLON: continue (Y or N)?

Ook hier wordt de Save beëindigd door elke toets anders dan "Y" en wordt u door de cursor op de fout gewezen. Deze kunt u dan gemakkelijk verbeteren, waarna u het bestand weer opslaat.

U zult zien dat u de bestanden die ZIP voor u schrijft meestal niet eens hoeft te bekijken, tenzij u codes wilt inlassen voor foutzoeken, enz. Breng uw applicatieprogramma op het scherm en voeg hieraan toe:

DO < naam >

of

SET FORMAT TO < naam >

en voer het uit. Als uw programma aan de variabelen die u met ZIP heeft ingevoerd waarden toekent, verschijnt het scherm of formulier geheel naar wens.

Om een nieuw bestand te openen, wist u eerst het vorige bestand met:

/E

en antwoordt u "Y" op de vraag:

Erase everything (Yes or No)?

Dan vraagt ZIP of u een nieuw of een bestaand bestand wilt hebben. *Als u wilt eindigen*, typt u Q bij de volgende boodschap:

< NEW > or < OLD > file (Q to QUIT)?

U beantwoordt de volgende vraag met "Y":

QUIT to system (Yes or No)?

Wanneer u *in* een scherm aan het werk bent, kunt u natuurlijk ook op ieder ogenblik eindigen, door gewoon:

/Q

in te toetsen.

Ook hier weer geeft ZIP een bevestiging dat u wilt eindigen voordat het alles gaat wissen en de bestanden gaat sluiten.

Als u bestand langer is dan het scherm, geeft ZIP u het volledige bestand nog eens snel te zien alvorens terug te keren naar uw besturingssysteem.

NOOIT MEER @..SAY..GET.

Nu gaan we ZIP gebruiken om enkele ..SAY- en GET-statements te schrijven. Dit gaat nu gemakkelijker dan ooit!

Wellicht wilt u aan een echt formulier of inputscherm beginnen, omdat u na dit hoofdstuk een bestand wilt uitvoeren.

ZIP kiest voor de Paginalengte automatisch één scherm wanneer u een nieuw bestand opent. Als u een eerder aangemaakt bestand laadt, stelt ZIP de paginalengte op de lengte van dat bestand in.

Als u het bestand wilt gebruiken voor een printformulier, kiest u een willekeurige Paginalengte (maximaal 88 regels).

Met de rij- en kolompositie die u in de ZIP Talker kunt aflezen, kunt u de titel, datum, commentaar en variabelen correct positioneren.

Om een variabele te positioneren, typt u het symbool @ (SAY een variabele) of het symbool # (GET een variabele) op de gewenste beginpositie voor het veld.

Nu toetst u de naam van de variabele in. U hoeft geen "einde-naam" teken te typen omdat ZIP de regels voor dBASE II variabelenamen kent. ZIP breekt de naam af wanneer het 10 geldige tekens gelezen heeft of wanneer het een teken aantreft dat niet geldig is in dBASE II (ook spaties). Als uw variabele 10 tekens lang is, mag u de positie vlak naast het laatste teken direct gebruiken zonder scheidingstekens tussen de variabelenaam en de rest van de regel.

dBASE II accepteert de volgende tekens in een variabelenaam:

A-Z a-z 0123456789 en een *ingesloten* dubbele punt (:)

Bij de invoer van de variabelenaam kunt u de dubbele punt overal typen, ZIP zal echter bij verkeerd gebruikte dubbele punten stoppen als het bestand < naam >.CMD geschreven wordt. U krijgt dan de gelegenheid om de fout te corrigeren (zie het hoofdstuk Save hierboven).

Als de variabelenaam een teken bevat dat niet geldig is in dBASE II, breekt ZIP de naam af en neemt aan dat u de volgende boodschap of titel met het ongeldige teken wilt laten beginnen.

U vraagt nu het Hulpscherm nog eens op door twee keer het commando-symbool te typen (//). U zult de commando's /D (of de < delete > / < rubout > -toets), /I, /K en /A wellicht goed kunnen gebruiken voordat u het bestand af heeft. U kunt ook /C gebruiken om tekst in het midden van de regel te plaatsen (centreren).

IN FORMULIERBESTANDEN OPGENOMEN dBASE II-COMMANDO'S

Om de behoefte aan een externe editor te beperken, laat ZIP u zoveel dBASE II-commando's in uw formulieren opnemen als waarvoor plaats is.

Het gebruik van dBASE II commando's is echter aan de volgende regels *gebonden*:

1. Het bevel moet tussen vierkante haakjes staan ([]).
2. Beide haakjes moeten op dezelfde regel staan (gebruik de dBASE II punt komma voor langere bevelen).
3. Commando's tussen haakjes moeten gescheiden zijn van tekst en andere commando's tussen haakjes.

Als scheidingstekens gebruikt u één commandoteken vlak vóór het linker haakje, of twee commandotekens (@@, ##, @# of #@). Commando's tussen vierkante haakjes hoeven niet gescheiden te zijn van voorafgaande variabelenamen.

Het commandobestand op de volgende pagina werd door ZIP geschreven uitgaande van de invoer (regel 35 en 37 buiten het scherm). De schermen illustreren de toepassing van deze regels.

20 ... ZIP

```
*B.CMD
SET FORMAT TO PRINT
SET MARGIN TO 40
RESTORE FROM B:CONSTANT
@ 7,5 SAY "*****"
@ 8,7 SAY "VOORBEELDFAKTUUR"
EEN ANDER dBASE II COMMANDO
@ 9,5 SAY "*****"
@ 12,5 SAY 'FAKTUUR'
@ 12,13 SAY faktnr
SAVE TO A
@ 12,40 SAY "DATUM"
@ 12,45 SAY datum
DO datumtest
SET FORMAT TO SCREEN
@ 15,6 SAY "KLANT"
@ 15,13 SAY klant
DO controle
@ 17,0 SAY "-----"
@ 17,50 SAY "-----"
USE B:KOSTEN
STORE 0 TO Totaal
DO WHILE .NOT. EOF
? Job:Nr,Omschr,Belastbaar
STORE TOTAAL + BELASTBAAR TO TOTAAL
ENDDO
@ 35, 0 SAY "-----"
@ 35,50 SAY "-----"
@ 37,41 SAY "TOTAAL FAKTUUR"
@ 37,57 SAY Totaal
SET FORMAT TO SCREEN
RETURN
```

VRAAG: *Hoe moet ik resultaten van expressies of verschillende veldinhouden vlak bij elkaar afbeelden als hun namen op de posities staan waar ik ze wil afbeelden?*

ANTWOORD: Als de mogelijkheid om dBASE II-commando's in uw bestand op te nemen geen uitkomst biedt, dan is dit één van de gevallen waarin u het door ZIP geschreven bestand < naam > .CMD naderhand nog zult moeten editen.

Maar ZIP maakt dit alles toch veel gemakkelijker omdat ZIP tot 40 GET- en SAY-symbolen op één regel van 80 tekens kan verwerken.

Eén van de toepassingen van deze mogelijkheid is dat alleen *symbolen* ingevoerd kunnen worden op de positie waar u korte velden wilt afbeelden of invoeren. Als de ZIP Talker "NO VARIABLE – continue (Y or N)? vraagt, typt u dan "Y".

Het voordeel van deze methode is, dat u in één oogopslag kunt zien waar de uitdrukkingen of variabelenamen moeten komen omdat ZIP in het bestand < naam > .CMD het volgende geschreven heeft:

of @ xx,yy SAY (rest van de regel is leeg)
 @ xx,yy GET (rest van de regel is leeg)

U krijgt echter voortdurend boodschappen van de ZIP Talker (en u zult vele prompts moeten beantwoorden). Daarom vindt u het misschien makkelijker om gewoon afzonderlijke cijfers of letters in te voeren in ZIP en deze cijfers of letters dan door de juiste naam en uitdrukking te vervangen bij het editen van het bestand < naam > .CMD.

Als u klaar bent met uw scherm of formulier, slaat u het bestand op met /S en typt u /Q om het programma te verlaten.

Opgelet:

Probeer niet met het schermbeeldbestand < naam > .ZIP te gaan werken. Het is een bestand in ASCII-code, wat betekent dat u het met uw CP/M-functie "TYPE" kunt inkijken (gebruik < control > -S om de scrolling-functie in- en uit te schakelen). Door de gebruikte opslagmethode echter zal dit bestand door een tekst-editor waarschijnlijk vernietigd worden.

Als u een afdruk van het scherm wenst, dan gebruikt u een tekstverwerker om het bestand < naam > .ZPR af te drukken, of u typt < control > -P en gebruikt het CP/M commando TYPE.

U kunt uw tekstverwerker gebruiken om de dBASE II-commandobestanden < naam > .CMD of < naam > .FMT te bekijken die door ZIP beschreven zijn. U kunt ook het CP/M-commando "TYPE" gebruiken om het snel te doorlopen voordat u het uit gaat voeren.

Start dBASE op en ga na of waarden toegekend zijn aan de variabelen die u in uw formulier geschreven heeft. Dan typt u:

of DO < naam >
 SET FORMAT TO < naam >

ANDERE BESTANDSTYPEN, OPMAAK

Normaliter werkt ZIP met een bestand < naam > .ZIP dat één onafgebroken reeks is, een beeld van het format dat u in het geheugen van de computer heeft aangemaakt. (Gelijksoortig en uitwisselbaar met de bestanden die IBM in één van zijn systemen gebruikt.)

De beperking van een *.ZIP bestand is dat het alleen met ZIP snel kan worden gewijzigd wat anders heel wat moeite kost.

Om dit probleem te omzeilen, kan ZIP ook elk standaard ASCII-bestand laden. Hiertoe vermeldt u gewoon een uitbreiding in de bestandsnaam. *GEBRUIK ECHTER NOOIT ".ZIP"* als uitbreiding voor een ASCII-bestand!!!

Deze functie kan gebruikt worden om schermen te laden die met WordStar, Magic Wand of een andere tekstverwerker opgesteld zijn. De beperkingen zijn dat elke regel op een < return > moet eindigen en dat het bestand niet meer dan 88 regels lang mag zijn.

Bovendien kunt u met deze functie de < naam > .CMD-bestanden wijzigen die door ZIP zijn geschreven. Zulke bestanden zijn gewoonlijk vrij kort en passen dan ook binnen de 88 beschikbare regels.

De ingesloten commando's die ZIP accepteert zijn handig maar kunnen in sommige gevallen (zoals in bepaalde lussen) in de "verkeerde" volgorde worden uitgevoerd.

Om dit te verhelpen, stelt u eerst uw rapportformat op met ZIP, dan slaat u dit format op en laadt u het commandobestand < naam > .CMD. Dan gebruikt u het commando /A om het nodige aantal blanco rijen in te voegen voor de DO WHILE...ENDDO-lus om toegang te krijgen tot de database en het rapport af te drukken. Deze lus vergt normaliter niet méér dan twee of drie extra regels.

Dan slaat u het format weer op, en de dBASE II-code voor de lus staat op de juiste plaats.

U zult zeker nog andere toepassingen kunnen bedenken voor deze mogelijkheid.

VOOR dBASE II:

geeft @ de positie aan voor "SAY een variabele", en
geeft # de positie aan voor "GET een variabele".

U hoeft geen "einde-veld"-teken te gebruiken, aangezien ZIP veronderstelt dat de variabelenaam compleet is bij het eerste teken dat daarvoor niet geldig is in dBASE II (max. 10 tekens). De geldige tekens voor dBASE II variabelenamen zijn:

A-Z a-z 0123456789 en een ingesloten dubbele punt (:)

ZIP controleert de printbestanden op niet-ingesloten dubbele punten, ontbrekende variabelenamen en GET's, en biedt u dan de gelegenheid om ze te verbeteren of te negeren. Om de 64 GET's schrijft ZIP een READ met bijhorende markering in het .CMD bestand, en brengt het een boodschap op het scherm om u te laten weten waar de READ zich bevindt.

ZIP schrijft commandobestanden (< naam > .CMD en < naam > .FMT) die u zonder verder editen kunt uitvoeren. In een schermbestand schrijft ZIP een ERASE aan het begin van het bestand. In printbestanden, schrijft ZIP SET FORMAT TO PRINT en SET MARGIN TO xx aan het begin van het bestand en SET FORMAT TO SCREEN vlak vóór de laatste RETURN.

ZIP slaat eveneens een beeld van uw scherm op als < naam > .ZIP en een afdrukbare kopie als < naam > .ZPR.

ZIP kan eveneens gebruikt worden voor het editen van standaard ASCII bestanden (max. 88 regels).

Vierkante haakjes ([]) zijn *GERESERVEERD* als scheidingstekens: ze mogen alleen gebruikt worden voor dBASE II-commando's en commentaar dat u in uw formulieren wilt opnemen.

DYNAMISCHE WAARDEN:

- is ingesteld als **Verticaal-symbool**, maar kan gewijzigd worden in "|" of een willekeurig ander teken tijdens de installatie van ZIP. Het kan ook tijdelijk veranderd worden tijdens de aanmaak van een formulier met ZIP.
- is het **Horizontaal-symbool** dat tijdens de installatie kan worden gewijzigd, en (net als het Verticaal symbool) te allen tijde veranderd kan worden tijdens het gebruik van ZIP.

TAB-AFSTAND is ingesteld op 5 spaties, maar kan tijdens het gebruik van ZIP veranderd worden in elke gewenste waarde tussen 1 en de scherm breedte min één.

PAGINALENGTE kan tijdens het gebruik van ZIP te allen tijde ingesteld worden op een willekeurige waarde van één scherm en 88 regels.

PRINTERMARGE mag ingesteld worden op een willekeurige waarde tussen 0 en 132.

ANDERE ZIP-COMMANDO'S EN SYMBOLEN:

- / is het **Commandosymbool**, maar moet geïnstalleerd worden als “#” of een ander teken dat u niet als tekst in uw schermen en uw output formulieren gebruikt.
- //: als u tijdens een sessie **twee** keer het commandosymbool intoetst, toont ZIP een samenvatting van de commando's en geeft u de gelegenheid om het Horizontaal-symbool, het Verticaal-symbool, de Tab-afstand, de Paginalengte en de Printer marge te wijzigen. Om bij voorbeeld de Paginalengte in 47 te veranderen, typt u:

```
//  
P  
47
```

Om de onderstaande commando's te gebruiken, typt u het geïnstalleerde commandosymbool gevolgd door het gewenste teken.

H is het commando voor het TREKKEN/WISSEN van **Horizontale lijnen**. Om dit commando te gebruiken, typt u:

```
/H
```

Als de cursor op een ander teken staat dan het teken dat u als Horizontaal-symbool gebruikt, dan wordt d.m.v. dit commando een lijn getrokken vanaf de cursorpositie tot aan de rechterraand van het scherm. Als de cursor op het teken staat dat als Horizontaal-symbool wordt gebruikt, worden alle tekens gewist vanaf de cursorpositie tot aan de rechterraand van het scherm.

V is het commando voor het TREKKEN/WISSEN van **Verticale lijnen**. Om dit commando te gebruiken, typt u:

```
/V
```

Als de cursor op een ander teken staat dan het teken dat u als Verticaal-symbool gebruikt, dan wordt d.m.v. dit commando een lijn getrokken vanaf de cursorpositie tot aan de onderste regel van uw pagina. Als uw pagina meer dan 23 regels lang is, loopt de lijn buiten het scherm door. Als de cursor op het teken staat dat als Verticaal-symbool wordt gebruikt, worden alle tekens gewist vanaf de cursorpositie tot de onderste regel van uw pagina.

ZIP plaatst en wist het “+” symbool automatisch wanneer de horizontale en de verticale lijnen elkaar snijden (alleen de huidige waarden – zie “Gebruik van ZIP”). ZIP houdt deze symbolen bij zodat u snel en makkelijk kaders kunt tekenen en wissen en zelfs ingewikkelder vormen kunt maken.

/H,/H en /V,/V kunnen worden gebruikt om respectievelijk rijen en kolommen te wissen.

T en B (voorafgegaan door het commandosymbool) brengen de cursor respectievelijk **bovenaan** (Top) en **onderaan** (Bottom) het scherm, in dezelfde kolom.

M brengt de cursor in het **midden van de rij** wanneer het bevel de eerste keer wordt ingevoerd, bij de tweede keer verspringt hij naar het midden van de scherm.

< tab > brengt de cursor eerst naar de rechterrاند van het scherm, waarna de cursor van de ene rand naar de andere springt.

C **centreert de tekst** in de regel die door de cursor wordt aangeduid.

N toont het **volgende scherm** (Next) en doorloopt het werkbestand in stappen van één scherm tot de laatste regel van uw pagina onderaan het scherm staat.

P toont het **vorige scherm** (Previous) en bladert terug in het werkbestand tot Rij 0 bovenaan het scherm staat.

F toont het **eerste scherm** (First), een manier om snel terug te keren naar het begin.

L toont het **laatste scherm** (Last), een manier om snel naar het einde van het formulier te gaan.

I **last een spatie in** (Insert) op de cursorpositie. Als uiterst recht in het scherm tekens verdwijnen, gaan deze tekens voorgoed verloren.

D **wist** (net als uw Delete- of Rubouttoets) **het teken** (Delete) waarop de cursor staat.

A **voegt een rij of kolom in** (Add) ter hoogte van de cursor. Als daarmee de ingestelde Paginalengte overschreden wordt, zodat de tekst gedeeltelijk van het scherm verdwijnt, gaat die tekst voorgoed verloren.

K **wist de rij of kolom** (Kill) waarin de cursor staat en laat de tekst aansluiten.

E **wist uw werkbestand** (Erase) zodat u aan een nieuw bestand kunt beginnen.

S **slaat uw werkbestand op** (Save) als < naam > .ZIP en < naam > .ZPR en **schrijft daarna het commandobestand** voor uw scherm of printout als < naam > .CMD of < naam > .FMT (naar keuze).

Q **verlaat het programma** (Quit) en keert terug naar het besturingssysteem.



SED

ASHTON · TATE ■TM

© 1985 Ashton-Tate
Reproduction by any method is strictly prohibited

Screen Editor

Een dBASE II R Generator van programmabestanden

De handleiding bij Screen Editor (SED) staat op de Screen Editor diskette; u kunt dus altijd informatie opvragen – ook wanneer u met Screen Editor een format voor uw rapport of voor een invoerscherm aan het genereren bent. Op de volgende bladzijden zal u duidelijk worden wat Screen Editor kan, wat u moet doen om met Screen Editor aan het werk te gaan, en wat u kunt doen bij problemen.

Wat is Screen Editor?

Screen Editor is een programma waarmee u op een eenvoudige manier schermen voor dBASE II kunt genereren. Met Screen Editor kan een dBASE II programmeur schermen genereren in een fractie van de tijd en met veel minder moeite dan voorheen. Aangezien Screen Editor met dBASE II werkt, moet u wel reeds vertrouwd zijn met dBASE II voordat u dit programma gaat gebruiken.

Screen Editor genereert automatisch dBASE II commandobestanden uitgaande van de schermen die u met Screen Editor aanmaakt. U “tekent” het scherm met behulp van de scherm-editor van Screen Editor en daarna geeft u Screen Editor gewoon opdracht om het dBASE II commandobestand te genereren. U kunt zelfs de meeste tekstverwerkingsprogramma's gebruiken om het scherm samen te stellen en het vervolgens naar Screen Editor overdragen om de dBASE II commando's te genereren.

Ook kunt u met Screen Editor rapportformats voor DBASE II genereren, hetgeen in grote mate overeenkomt met het genereren van schermen. Rapportformats genereren met behulp van Screen Editor is bijzonder handig omdat dan de tekst en de input- en outputvelden in de correcte volgorde geschreven worden. Is het rapport breder dan het scherm, dan kunt u het rapportformat altijd nog uittekenen. Met Screen Editor kunt u immers in horizontale en verticale richting door het scherm rollen.

Screen Editor is een krachtige full-screen editor, die ook geschikt is voor de aan- en opmaak van andere bestanden.

2 ... SED

Eigenschappen van Screen Editor

- Met Screen Editor genereert u moeiteloos schermen en rapportformats voor dBASE II.
- De Screen Editor handleiding is steeds on-line beschikbaar.
- De handleiding bevat een volledige inhoudsopgave, een index en bijlagen.
- Elke pagina van deze handleiding kunt u met een eenvoudig commando afdrukken.
- Screen Editor's editor biedt uitgebreide mogelijkheden voor cursorbesturing.
- De editor kan de schermen zowel horizontaal als verticaal over het scherm rollen.
- De editor beschikt over tal van commando's om tekst te verplaatsen, te kopiëren of te wissen, hetgeen zeker voor schermen met tekst in kolommen of in rechthoekige vorm erg handig is.
- De commando's van de editor zijn volledig aan te passen. Mochten de toetsen die wij voor de editor-commando's gekozen hebben u minder geschikt lijken, dan kunt u ze door iedere willekeurige toets vervangen. Als u speciale toetsen op uw toetsenbord wil gebruiken, dan kan dit zonder enig probleem (zie onderstaande beschrijving van SINSTALL).

Installatie van Screen Editor: het SINSTALL programma

Het doel van SINSTALL is tweevoudig:

- (1) Screen Editor aanpassen aan uw computer en uw scherm, en
- (2) Screen Editor aanpassen aan uw toetsenbord en de edit-commando's van uw voorkeur.

Als Screen Editor door uw dealer geïnstalleerd is, kunt u dit hoofdstuk overslaan, tenzij u enkele standaard-systeemwaarden of voorgeprogrammeerde toetsen wilt veranderen.

Opmerking voor CP/M-86 gebruikers: Vervang "EXE" overal in dit hoofdstuk door "CMD".

Aanpassing van Screen Editor aan uw computer en uw scherm

De dealer bij wie u Screen Editor gekocht heeft, zal het programma normaliter aan uw computer en scherm hebben aangepast. Wilt u echter een wijziging aanbrengen, dan kunt u dit zelf doen of het door uw dealer laten verzorgen. Als u een IBM computer met een Monochrome of een Color-Graphics adapter gebruikt en de respons van Screen Editor is niet correct, neemt u dan contact op met uw dealer – een speciale versie van Screen Editor is voor deze adapters beschikbaar.

Als uw de aanpassing zelf gaat uitvoeren, dan heeft u daarbij de handleiding van uw terminal nodig (of de handleiding van uw scherm, wanneer uw scherm van een ander merk is dan het toetsenbord). In deze handleiding vindt u de speciale codes die door de terminal herkend en geïnterpreteerd worden als instructies om de cursor of de weergave te besturen. Deze codes worden doorgaans “escape sequences”, “terminalbesturingscodes” of “terminal attribuutcodes” genoemd. Als zulke codes niet vermeld zijn, raadpleegt u voordat u verder gaat eerst even uw dealer.

In dit document gebruiken we de term “terminal attribute sequence” voor al deze speciale codes.

Voor de positionering van de cursor op het scherm is een terminal attribute sequence vereist. Screen Editor accepteert twee soorten cursorpositionering. Bij de eerste soort worden de cursorcoördinaten voorgesteld door twee ASCII-waarden, gescheiden door een puntkomma (;). De tweede soort cursorpositionering gebruikt twee opeenvolgende bytewaarden voor de cursorcoördinaten.

ANSI terminals

Als de terminal met de eerste cursorpositionering werkt, dan gaat het om een ANSI terminal. Screen Editor accepteert iedere willekeurige codereeks die aan de waarden van de cursorcoördinaten voorafgaat en iedere willekeurige bytewaarde die op de coördinaatwaarden volgt. De eerste codereeks (die aan de coördinaatwaarden voorafgaat) noemen we de “aanhef cursorpositionering”. De tweede code (de byte na de coördinaatwaarden) noemen we de “afsluitingsbyte”. U moet beide codes intoetsen wanneer SINSTALL u hierom vraagt.

Voorbeeld:

In de terminalhandleiding leest u dat voor de cursorpositionering de volgende code vereist is.

```
ESC [# ; #H
```

... de #-tekens staan voor de rij- en de kolomcoördinaten.

De aanhef van de cursorpositie is “ESC [” en de afsluitingsbyte is “H”. (“ESC” slaat op de code die door de Escape-toets wordt gegenereerd en niet op de afzonderlijke letters “E”, “S” en “C”).

Niet-ANSI terminals

Naast de codes van de ANSI-conventie, accepteert Screen Editor ook twee opeenvolgende bytewaarden als cursorcoördinaten. In Screen Editor mogen de coördinaatbytes door een willekeurige code sequence voorafgegaan worden en SINSTALL vraagt deze code in de boodschap "cursor positioning lead-in" (aanhef cursorpositionering). Screen Editor biedt bovendien de mogelijkheid om de eigenlijke schermcoördinaten te berekenen door een willekeurig constant getal tussen 0-210 af te trekken van de coördinaatbytes. SINSTALL vraagt deze constante in de boodschap "coordinate offset (default = 32)".

Voorbeeld:

Uw terminalhandleiding geeft aan dat de cursor gepositioneerd wordt met de volgende code:

`ESC = rc`

... waarbij *r* voor de rij- en *c* voor de kolomcoördinaat staat.

De aanhef van de cursorpositionering is "ESC="". Om de constante te bepalen die bij de waarden *r* en *c* moet worden opgeteld, kijkt u of de waarden *r* en *c* gelijk zijn aan het rij- en kolomnummer (bovenste rij en de uiterst linkse kolom = 0) of om te kijken of een constante van deze waarden moet worden afgetrokken om het reële rij- en kolomnummer te krijgen (bovenste rij is rij 0, uiterst linkse kolom is kolom 0). De meeste constructeurs leveren een volledige tabel van alle mogelijke waarden van de rij- en kolombytes. Als u over een dergelijke tabel beschikt, dan loopt u die even door om te zien of de waarden met 1 stijgen als het rij- of het kolomnummer met 1 toeneemt. Als dit zo is, is het eerste getal in de tabel de coördinaatafwijking.

Screen Editor Standaardwaarden

We hebben de cursor-positionerings-sequence het eerst besproken, ook al wordt deze informatie pas in de derde sectie opgevraagd. Terminals die aan de ANSI-conventie beantwoorden en terminals die niet tot het ANSI-type behoren werden reeds behandeld, evenals het eindteken voor ANSI-terminals en de coördinaatafwijking voor niet-ANSI terminals.

SINSTALL zal u de rest van de gegevens voor deze sectie opvragen via een reeks prompts waarvoor u de standaardwaarden al dan niet kunt accepteren. Het gaat om vragen als:

- Do you want INSERT or OVERWRITE
- Default to continuous display of current cursor positioning in the status line?
- Should the status line be at the top or the bottom of the screen?
- How many rows does your terminal display?
- How many columns does your terminal display?
- The minimum form width you require for all forms. Screen Editor may make the form width greater than this number but never less.

Sectie toetsenbord-timing

Deze sectie levert één getal aan Screen Editor. Dit getal is de geschatte vertraging tussen toetsaanslagen wanneer uw toetsenbord over een automatische repeteerfunctie beschikt. Screen Editor heeft deze geschatte waarde nodig om een onderscheid te kunnen maken tussen twee of meer toetsen die na elkaar worden ingedrukt en de overdracht door het toetsenbord van meervoudige codes in een "burst". Deze waarde hangt af van alle volgende factoren; als één van deze factoren verandert, moet u Screen Editor opnieuw installeren en de nieuwe vertragingfactor berekenen:

- (a) het toetsenbord
- (b) cpu- en bussnelheid
- (c) monitor interface
- (d) het besturingssysteem

Monochroom of kleurenscherm

Het installatieprogramma brengt de volgende prompt op het scherm:

"Do you have the Monochrome or Color/graphics screen (m/c)?"

Typ M voor een Monochrome-adapterkaart of C voor een Color/graphics-adapterkaart.

Sectie Terminal attribute sequence

In deze sectie voert u alle terminal attribute sequences in die door Screen Editor gebruikt worden. De sequences mogen in hexadecimale of in ASCII-notering worden ingevoerd. Invoer in ASCII moet tussen aanhalingstekens worden geplaatst. De sequence "ESC="; kunt u invoeren als 1B"=" of: "<esc> =" ... waarbij <esc> betekent dat u de Escape-toets na het eerste aanhalingsteken indrukt.

Screen Editor zal de volgende codes vragen:

Screen clear...mandatory, Screen Editor will not run without it.
 Cursor positioning lead-in...compulsory
 Erase to end-of-line...highly desirable
 Add new line at cursor...highly desirable
 Switch off cursor...optional
 Switch on cursor...compulsory only if 'switch off' given.

Vervolgens vraagt het programma u tot 10 videocodes naar keuze in te vullen. Codes 0 en 1 moeten worden ingevoerd; de overige codes zijn optioneel, ze worden alleen gebruikt voor het genereren van commandobestanden door dBASE II. Door een code 0 wordt alle volgende tekst in de normale modus afgebeeld. Tekst die op een code 1 volgt, wordt afgebeeld in "video-negatief".

6 ... SED

Onderstaande lijst geeft een overzicht van de aanbevolen codes, met uitleg tussen haakjes. Instructies voor het invoeren van de codes voor de scherm display vindt u in de on-line handleiding.

- Code 0: Normaal (wit tegen zwarte achtergrond, hoge intensiteit)
- Code 1: (Lage intensiteit)
- Code 2: (Video-negatief)
- Code 3: (Onderstrepen)
- Code 4: (Knipperen)
- Code 5: (Negatief onderstrepen)
- Code 6: (Negatief knipperen)
- Code 7: (Knipperend onderstrepen)
- Code 8: (Negatief lage intensiteit)
- Code 9: (Knipperen lage intensiteit)

Daarna vult u de volgende codes in om eventuele codes die u in code 2 tot en met 9 heeft ingesteld, te annuleren.

- Code 10: (Forceert normale weergave)
- Code 11: (Forceert hoge intensiteit)
- Code 12: (Annuleert 2)
- Code 13: (Annuleert 3)
- Code 14: (Annuleert 4)
- Code 15: (Annuleert 5)
- Code 16: (Annuleert 6)
- Code 17: (Annuleert 7)
- Code 18: (Annuleert 8)
- Code 19: (Annuleert 9)

Aanpassing van het toetsenbord

De editor-commando's van Screen Editor kunnen door iedere willekeurige toets ingeschakeld worden. In de oorspronkelijke Screen Editor versie bijvoorbeeld kunt u met de toetsen ^E, ^S, ^D en ^X de cursor verplaatsen. Uw toetsenbord kan echter "cursorbesturingstoetsen" hebben die u liever gebruikt; met SINSTALL kunt u andere toetsen instellen. Dit is echt niet moeilijk, ondanks de vrij uitvoerige uitleg die u van SINSTALL krijgt.

Als u de toetsen wilt veranderen in uw versie van de Screen Editor, moet u, voor elk editor-commando van Screen Editor, drie vragen beantwoorden:

- Eén: Hoeveel toetsaanslagen wilt u in de nieuwe toetsenreeks hebben.
- Twee: SINSTALL zal u vragen deze toetsen langzaam in te drukken.
- Drie: SINSTALL vraagt een mnemonic voor deze toetsenreeks, als geheugensteuntje. Deze mnemonic zal in de on-line handleiding verschijnen.

Enkele tips om tijd te besparen

U kunt de antwoorden voor een vorig Screen Editor-commando aanhouden door een <return> in te voeren als antwoord op de eerste vraag (aantal toetsaanslagen). Zo behoudt SINSTALL de vorige toetsaanslagen en het mnemonic en stapt over naar de volgende commando. Als u de toetsaanslagen wilt behouden maar het mnemonic wil veranderen, dan typt u een spatie als antwoord op de eerste vraag, waarna u een nieuw mnemonic kunt invoeren.

U typt het mnemonic zo snel u wil, maar u mag de eigenlijke toetsaanslagen niet sneller invoeren dan het toetsenbord een toets automatisch kan repeteren. Als u te snel typt, neemt SINSTALL aan dat u slechts één toets heeft ingedrukt, ook als u twee toetsen na elkaar heeft ingedrukt.

Als u klaar bent met het laatste commando geeft SINSTALL u een tweede kans en vraagt of u alsnog de codes wil veranderen.

**Uitvoering van SINSTALL
SED.EXE en bibliotheken**

INSTALL kan twee soorten bestanden uitlezen en bijwerken: SED.EXE; of een bibliotheekbestand met aanpassingsgegevens. SINSTALL herkent elk bestand van het type .EXE (.CMD in CP/M-86) als een versie van Screen Editor. Als u SED.EXE wilt veranderen, dan moet u alle oorspronkelijke initialiseringsgegevens van SED.EXE uitlezen en ze in een bibliotheek opslaan voordat u wijzigingen aanbrengt. Als er iets verkeerd gaat in de aanpassing, volstaat een herinitialisering van SED.EXE met de oorspronkelijke gegevens. Er van uitgaand dat u deze bibliotheek MY.LIB noemt, gebruikt u het volgende commando:

```
INSTALL SED.EXE MY.LIB/0
```

De “/0” die op “MY.LIB” volgt, is het bibliotheeknummer dat toegekend werd voor de opslag van de initialiseringsgegevens van Screen Editor. Als u Screen Editor nogmaals wilt initialiseren, dan toetst u het volgende in:

```
INSTALL MY.LIB/0 SED.EXE
```

Om een bestaande initialisering te wijzigen, typt u:

```
INSTALL MY.LIB/n ...   n is het bibliotheeknummer van de bestaande
                       gegevens, of,
INSTALL PROG.EXE ...   PROG.EXE is een versie van Screen Editor.
```

Dan loopt SINSTALL alle hierboven beschreven secties door en vraagt bij elke sectie of u iets wilt veranderen. Als u klaar bent, vraagt SINSTALL of u de wijzigingen wilt opslaan. Als dit het geval is, biedt SINSTALL u de gelegenheid om de gewijzigde gegevens in een ander bestand en/of onder een ander bibliotheeknummer op te slaan. Als u een nieuwe bibliotheek wilt openen met volledig nieuwe informatie, dan typt u:

```
INSTALL NEW.LIB/0
```

en SINSTALL zal u vragen of uw bestand nieuw is. Als u “Y” (“Yes”) als antwoord intoetst, zal SINSTALL het bestand openen.

Reserve-kopie van uw Screen Editor software

Gebruik voor normale bewerkingen nooit de originele kopie van Screen Editor, maar altijd een reserve-kopie en berg de oorspronkelijke diskette goed op, zodat u die indien nodig opnieuw kunt kopiëren.

Op de diskette met het label "Sample Programs Disk" staat alle Screen Editor software. Maak allereerst een kopie van deze software op een geformatteerde diskette en gebruik voortaan de kopie van Screen Editor. Bewaar de originele diskette op een veilige, koele en droge plaats, niet in de buurt van elektrische apparatuur die magnetische velden kan veroorzaken. Screen Editor en dBASE II zijn gepatenteerde software-produkten waarvoor een copyright geldt. U mag Screen Editor alleen voor persoonlijk gebruik op uw computer kopiëren. De licentie-overeenkomst die bij de diskette zit, verschaft hierover nadere informatie.

U heeft twee (of meer) diskette-drives:

Plaats de SED diskette in drive A en een geformatteerde diskette in drive B.

Bij gebruik van het CP/M besturingssysteem, typt u het volgende commando:

```
PIP B:=A:.* [V] <return >
```

Bij gebruik van het MS-DOS of PC-DOS besturingssysteem, typt u:

```
COPY A:.* B:/V <return >
```

Heeft u één diskette-drive en één Winchester-schijf:

Eerst kent u een (letter)code toe aan de diskette-drive en de Winchester drive. In ons voorbeeld gebruiken we "F" als code voor de diskette drive en "W" voor de Winchester-drive. (Gebruikt u een IBM XT, dan vervangt u de codes "F" en "W" respectievelijk door "A" en "C".)

CP/M-gebruikers

Plaats de Screen Editor diskette in de diskette-eenheid en typ het volgende commando:

```
PIP W:=F:.* [V] <return > (typ de passende drive codes in plaats van F en W)
```

Na dit commando plaatst u de blanco diskette in de diskette-drive en typt u:

```
PIP F:=W:.* [V] <return >
```

Uw back-up is klaar; bovenstaande commando's hebben de Screen Editor software eveneens op uw Winchester-schijf gekopieerd.

MS-DOS of PC-DOS gebruikers

Schuif de Screen Editor-diskette in de diskette-eenheid en typ het volgende commando:

DCOPY F:*.* W:/V <return> (typ passende drive codes i.p.v. F en W).

Als het proces voltooid is, plaatst u de blanco diskette in de diskette-drive en typt u:

COPY W:*.* F:/V <return>

U heeft slecht één diskette-drive:**CP/M-gebruikers**

Zoek het commando voor het kopiëren van een diskette in uw handleiding op en volgt de instructie. Uw CP/M handleiding zal het over source-diskettes en destination- of target-diskettes hebben. De source-diskette is de diskette waarop de Screen Editor software staat en de destination- of target-diskette is de blanco diskette waarop u gaat kopiëren.

MS-DOS- en PC-DOS-gebruikers

Steek de DOS diskette in de schijfeenheid en typ het volgende commando:

DISKCOPY A:

Verwijder de DOS diskette en breng de Screen Editor-diskette in wanneer u daarom d.m.v. een prompt gevraagd wordt. Als het commando DISKCOPY de Screen Editor software zo ver mogelijk uitgelezen heeft, brengt u de target-diskette in, dit is de blanco geformatteerde diskette. Het commando DISKCOPY schrijft alles weg wat tot nu toe van de geformatteerde diskette uitgelezen werd. Het is mogelijk dat DISKCOPY nog een prompt geeft waarin gevraagd wordt de Screen Editor-diskette nogmaals in te brengen. Herhaal de procedure tot alles gekopieerd is.

Aan de slag met Screen Editor

Zodra u uw Screen Editor-software gekopieerd heeft en de originele diskette op een veilige plaats heeft opgeborgen, wilt u Screen Editor wellicht op uw dBASE werkdiskette kopiëren. Steek dan de werkdiskette in de drive en als de prompt van het besturingssysteem verschijnt typt u SED gevolgd door een <return> .

Screen Editor

Als er niets gebeurt drukt u op < return >. Als er dan nog niets gebeurt, probeert u het met de oorspronkelijke Screen Editor-diskette nog eens. Als er ook dan niets gebeurt, controleert u op het diskettelabel of u de correcte versie voor uw computer heeft. Als dat zo is, is de diskette beschadigd, of heeft u een Screen Editor-versie gekregen die voor een andere computer bestemd is. Neem dan contact op met uw dealer.

Als alles goed gaat, komt het menu van Screen Editor op scherm. Dat ziet eruit als een functioneel blokdiagram dat aangeeft welke commando's u voor de verschillende taken moet invoeren.

Als u een verwarde tekst of rare tekens op het scherm ziet, dan is uw versie van Screen Editor niet geschikt voor uw terminal. Bel uw dealer en geef het merk en het model van uw terminal en uw toetsenbord op. Als u denkt het probleem zelf te kunnen verhelpen, neemt u de handleiding bij uw terminal en met het programma SINSTALL verbetert u de fout zelf. U moet echter eerst bepalen welke escape sequences uw terminal als speciale commando's nodig heeft en vervolgens deze sequences in SINSTALL invoeren.

Indien alles naar wens verloopt, typt u "?" wanneer het menu verschijnt. U komt dan in de inhoudsopgave van de on-line handleiding van Screen Editor. Als u niet vertrouwd bent met programma's als Screen Editor, is het beter de hele handleiding vanaf pagina 10 door te nemen. Ervaren programmeurs stappen wellicht meteen over naar de samenvatting van de commando's. Vergeet de INDEX niet, deze kan bijzonder goed van pas komen.

Als u deze on-line informatie onder de knie heeft, volgt u de instructies in de prompt onderaan de pagina en drukt u op < return > om terug te keren naar het hoofdmenu. We hebben een bestand gemaakt waarmee u alles kunt oefenen. Typ "E" en vul READ.ME in als het op te maken bestand; hierna moet het bestand READ.ME op uw scherm verschijnen. Als u vergeten bent wat u precies moet doen, kunt u op de volgende manier terugkeren naar de on-line instructiehandleiding. Kijk eerst even naar uw scherm: u ziet dat één regel op het scherm minder helder is dan de andere. Deze regel staat ofwel bovenaan ofwel onderaan het scherm. (Als alle regels er even helder uitzien, draait u even aan de contrastknop op de monitor). In deze zgn. "statusregel" staat zo ongeveer het volgende:

READ.ME OVERWRITE xxx for help

In plaats van OVERWRITE kan hier ook INSERTING staan, al naar gelang de Screen Editor-versie. Dit is een standaard instelling waarvan u kunt afwijken met de toets < insert > (of < ins >). In de OVERWRITE-modus brengt u de cursor naar de letter die u wilt veranderen en typt u de nieuwe letter over de andere heen. In de INSERT-modus voert u nieuwe letters in op de cursorpositie, waardoor alles naar rechts verschuift en er ruimte ontstaat voor uw input. Deze twee modi dienen vanzelfsprekend voor de opmaak van uw invoer op het scherm.

De toets die we met "xxx" hebben aangeduid, is de toets waarmee u terugkeert naar de on-line handleiding. In plaats van "xxx" ziet u een mnemonic voor één of meer toetsen die u indrukt. Als de mnemonic NOT CLEAR verschijnt, stapt u over naar het deel over de Screen Editor-conventies.

Nu drukt u de toets (of de toetsen) in die u naar de handleiding brengt/brengen. U krijgt dan de Inhoud van de on-line handleiding waaruit u een pagina met een bepaalde instructie kunt selecteren. Als u de informatie gelezen heeft, drukt u op < return > om terug te keren naar het READ.ME bestand. Op pagina 53 van de on-line handleiding vindt u de instructies voor een SAVE. U verlaat READ.ME zonder het bestand op te slaan door < esc > -Q te typen. Als u in READ.ME niets gewijzigd heeft, kiest u een optie. Als u "Q" typt, keert u terug naar uw besturingssysteem. Met een "?" keert u terug naar de Inhoud. Heeft u iets gewijzigd of iets toegevoegd aan READ.ME, dan vraagt het systeem of u het bestand wilt wissen zonder het eerst op te slaan. Typ "Y" en het scherm wordt gewist; het bestand READ.ME blijft echter ongewijzigd.

Gebruik van Screen Editor

Vergeet niet dat u te allen tijde de benodigde informatie in de on-line handleiding kunt opzoeken. Als de informatie die u zoekt niet in de Inhoud staat, kijkt u even in de Index. Ervaren gebruikers kunnen meteen overstappen naar de samenvatting van de commando's.

Een bepaalde pagina van de handleiding die u vaak nodig heeft, kunt u uitprinten en deze zo altijd bij de hand hebben. Wanneer u aan het eind van een pagina bent, volgt u de instructies onderaan het scherm voor het printen van de pagina. Als u slechts één bepaald scherm nodig heeft, gebruikt u de functie "print screen" van uw computer.

De handleiding bij Screen Editor is niet bedoeld als handleiding voor het programmeren van dBASE II. Gebruik voor vragen betreffende dBASE II uw dBASE II-handleiding. De Screen Editor-handleiding bevat echter voorbeelden van schermen en van de dBASE II-code die aan de hand van deze schermen wordt gegenereerd.

Het commando "G" in het hoofdmenu van Screen Editor zal op basis van uw schermbestand een dBASE II commandobestand genereren. Als u een bepaald bestand heeft opgemaakt of opgevraagd, neemt Screen Editor aan dat dit het bestand is waarvoor u een dBASE II commandobestand wilt hebben. Mocht u Screen Editor per ongeluk de instructie geven een commandobestand te genereren van een bestand dat niet op het scherm staat, dan zal Screen Editor waarschijnlijk allerlei foutmeldingen geven terwijl het uw opdracht probeert uit te voeren.

U kunt Screen Editor op twee manieren opstarten: vanuit uw besturings-systeem of 'Screen Editor' typen gevolgd door een bestandsnaam. Dit geeft hetzelfde resultaat als wanneer u naar het hoofdmenu overgaat, de optie "E" (Existing file) kiest en een bestandsnaam invoert.

12 ... SED

Screen Editor zal u af en toe vragen tekst in te voeren; dit kan een bestandsnaam, een paginanummer of een te zoeken tekstreeks zijn. Als u een fout typt, kunt u met de terugspatieertoets het teken links van de cursor wissen. Als u niets typt of de ingevoerde tekst wist, verlaat Screen Editor het geselecteerde commando. Screen Editor gebruikt de ingevoerde tekst wanneer u een <return> geeft.

Na het genereren van een commandobestand (.CMD of .PRG), kunt u het bekijken of opmaken met Screen Editor als ieder ander tekstbestand. U moet dan echter wel alle speciale dBASE II-commando's tussen vierkante haakjes [] in uw schermbestand opnemen. Als u dat niet doet, moet u dat iedere keer als u het bestand genereert, alsnog doen.

Conventies in Screen Editor

Bestandsspecificaties

Screen Editor past de vrijwel algemeen geldende conventies voor de specificatie van bestanden toe. Ter illustratie van deze afspraken enkele voorbeelden:

A:PAYBL.CMD

Betekent dat het bestand met de naam PAYBL op drive A staat en tot het type CMD behoort. De commando's mogen in hoofdletters en in kleine letters worden ingevoerd.

SCRATCH.DOC

Specificeert een bestand op de huidige drive. Het is opgeslagen onder de naam SCRATCH en het type is DOC.

De drive-codes (A, B, enz.) gevolgd door een dubbele punt (:) moeten vermeld worden, tenzij het bestand zich in de huidige drive bevindt.

Weergave van de toetsen

Uw terminal heeft een toets met het opschrift CONTROL, CTRL of CONT. Wanneer u bij het typen van een letter, deze <control>-toets ingedrukt houdt, genereert het toetsenbord een speciale code die de computer als een commando interpreteert in plaats van een gewone toetsaanslag. In Screen Editor wordt de CONTROL toets weergegeven door het symbool "" en ^Z betekent dan ook dat u de <control> en Z tegelijk moet indrukken.

Op uw toetsenbord ziet u ook een toets die Alt of ALT is genoemd; deze verkorting staat voor "alternate", wat betekent dat deze toets als alternatieve <control>-toets fungeert. Hij genereert andere commando's dan de <control>-toets wanneer dezelfde letter wordt gebruikt. <alt>-H betekent dat <alt> en H tegelijk moeten worden ingedrukt.

Sommige versies van Screen Editor gebruiken een speciale toets, <esc> of <escape>, voor de invoer van speciale commando's. Heeft u deze toets nodig, dan drukt u hem één keer in, net zoals de wagenterugkeertoets (<return>) of een gewone lettertoets. Houd deze toets echter niet ingedrukt omdat anders het toetsenbord een reeks escape codes naar de computer zendt. <esc>-D betekent "druk tegelijk op <esc> en op D".

Escape sequences

Screen Editor gebruikt de term "escape sequence" als benaming voor een speciale code die door de computer naar het scherm wordt gezonden en door de schermmonitor als een commando wordt beschouwd in plaats van weer te geven tekst. Zo kunt u er een escape sequence om bepaalde woorden met hoge intensiteit en andere woorden minder helder door de schermmonitor te laten afbeelden.