

MINISTERIE VAN LANDBOUW

Bestuur voor Landbouwkundig Onderzoek

Rijkscentrum voor Landbouwkundig Onderzoek - Gent

RIJKSSTATION VOOR ZEEVISSERIJ - OOSTENDE

Directeur : Dr. P. HOVART

Programma voor het beheer van literatuurgegevens

W. DESCHACHT



MINISTERIE VAN LANDBOUW

Bestuur voor Landbouwkundig Onderzoek

Rijkscentrum voor Landbouwkundig Onderzoek - Gent

RIJKSSTATION VOOR ZEEVISSERIJ - OOSTENDE

Directeur : Dr. P. HOVART

Programma voor het beheer van literatuurgegevens

W. DESCHACHT

PROGRAMMA VOOR HET BEHEER VAN

LITERATUURGEGEVENS.

W. DESCHACHT

INLEIDING.

De tekst valt in vier delen uiteen:

1. een inleiding waarin het doel en de middelen omschreven worden;
2. een tweede gedeelte waarin de mogelijkheden van het programma aan de hand van een aantal schermlayouts geschetst worden;
3. in een derde gedeelte wordt de samenhang tussen de verschillende procedures duidelijk gemaakt. Dit gebeurt aan de hand van lijnschema's. Ofschoon ze enige gelijkenis vertonen aan een stroomschema zijn ze dit niet. Ze kunnen eerder een programmaschets genoemd worden;
4. de verschillende modules uit de hierboven geciteerde programmaschets worden stuk voor stuk uitgewerkt in het vierde en laatste gedeelte.

Doel.

Het objectief is het ontwikkelen van een programma waarmee het op een eenvoudige wijze mogelijk is een groot bestand of een ruime verzameling aan records te raadplegen en dit zonder op een dure infrastructuur beroep te moeten doen. De gebruiker moet in staat zijn om snel een antwoord te verwerken op vragen in de aard van:

- waar vind ik iets over ...
- wat zit er in de voor mij direct toegankelijke bibliotheek aan gegevens in verband met ...

De middelen daartoe evolueerden sterk in het voorbije decennium. Grote instellingen bieden hun medewerkers nog altijd heel wat meer mogelijkheden dan kleinere, maar ook daar is de weinig efficiënte steekkaart of de al wat modernere randponskaart aan een vervanging toe. De proliferatie van de PC bracht de mogelijkheden van de elektronische steekkaart in vrijwel eenieders bereik. Stelt het opstarten van het traditionele kaartsysteem geen problemen, dan verloopt het van wal steken met de elektronische tegenhanger wel wat moeizamer. De hard- en de software zijn immers niet geheel en al vrij te kiezen. Het opbergen en terugzoeken van gegevens kan voor een niet IBM compatibele uitrusting, bv een McIntosh, met behulp van een programma als OMNIS gebeuren. Voor een IBM compatibele uitrusting zal gedacht worden aan dBASE, FoxBASE, DATAFLEX of dergelijke. Zo'n programma kan als zodanig gebruikt worden, maar het wordt allemaal gebruiksvriendelijker wanneer de opdrachten in een uitvoeringsprogramma opgenomen worden. Ook diegenen die geen of weinig belangstelling voor de informatica hebben en geen dikke handleidingen willen doorworstelen kunnen dan snel met de nieuwe techniek vertrouwd gemaakt worden. Het hierna beschreven programma werd in dBASEIII+ ontwikkeld en is te beschouwen als een model. Een relatief gemakkelijke concretisering van een concept dat, na op grote schaal getoetst en zo nodig aangepast te zijn door een compilatie in een nog bruikbaarere vorm kan gegoten worden.

Omgeving.

Het programma werd ontwikkeld op een PC-XT. Het toestel draaide onder MS-DOS, versie 3.2. De fundamentele idee was een zo groot mogelijk aantal steekkaarten direkt toegankelijk maken. Het programma werd dan ook geschreven in de optiek dat een harddisk tot de normale uitrusting behoort, maar het is ook bruikbaar wanneer slechts twee floppiedrives van 360 kb beschikbaar zijn. Bij de ontwikkeling werden het programma en het permanente gegevensbestand ingeschreven op de D: drive in de subdirectory BIB. Het toestel was verbonden met een kleine STAR LC-10 printer.

Steekkaart.

Teneinde een zo groot mogelijk aantal steekkaarten te kunnen onderbrengen in een direct toegankelijk bestand, werden de velden geconcipieerd vanuit het oogmerk van een RAADPLEGING. De veldinhoud is niet geschikt voor het opstellen van een referentielijst bij een artikel of bij het uitwerken van een monografie over een onderwerp of een persoon. Met het ontworpen programma zou zo'n opdracht het creëren vereisen van een nieuw bestand. Hierin moeten dan de geselecteerde steekkaarten gekopieerd worden. Met behulp van de dBASEIII+ mogelijkheden moeten vervolgens de veldbreedtes aangepast worden en tenslotte zou de veldinhoud op de gepaste wijze moeten aangevuld worden.

De gebruikte steekkaarten bestaan uit de volgende velden:

Veldnaam	Type	Aantal tekens
AUTEUR	Karakter	30
TITEL	Karakter	120
DRAGER	Karakter	80
PLAATS	Karakter	60
TREFWRD	Karakter	120
AARD	Karakter	1
MEMO	Memo	10

De recordlengte bedraagt 392 bytes. En vermits dBASEIII+ niet alle alfa-numerieke tekens verwerkt, accenten kunnen problemen veroorzaken, gebeurde de invoer ALTLJD in hoofdletters.

Velden.

AUTEUR:

Dit veld kan slechts 30 tekens bevatten. Het vermelden van iedere co-auteur wordt niet beoogd. De invoer wordt beperkt tot de eerste auteur, gevolgd door de eerste letter van de eerste voornaam, gevolgd door een punt. Indien er co-auteurs zijn kan de gebruiker hierop geattendeerd worden door de toevoeging ET AL.

TITEL:

Voor het invoeren van de titel zijn 120 tekens beschikbaar. Dit zal niet altijd volstaan. De gebruiker kan het ontoereikend zijn signaleren door op het einde enkele punten in het veld in te toetsen.

DRAGER:

Dit veld bevat de eigenlijke bibliografische gegevens. Voor een artikel uit een tijdschrift komt hier de, liefst algemeen aanvaarde, afkorting van de naam, het jaartal, het volume, het nr en het nr van de eerste pagina. Met boeken zijn er problemen. Een boek kan inderdaad als boek opgenomen worden, maar er kan evenzogoed een verwijzing naar een hoofdstuk of een pagina nodig zijn. In het eerste geval wordt hier de naam van de uitgever, de plaats van uitgave, het jaartal en het ISBNnr opgenomen. Gebeurt de referentie naar een gedeelte van een boek, dan komt in dit veld de titel van het boek, het jaartal en de pagina. In het bestand is er dan wel een tweede steekkaart waarop het boek als boek vermeld wordt. Het gebeurt dat niet voldoende plaats beschikbaar is om alles op te nemen. Bij het beoordelen van wat er ingetoetst wordt dient de gebruiker te bedenken dat dit een belangrijk indexveld is. Het zou ideaal zijn mochten de hierin opgenomen gegevens uniek zijn. Ondermeer in verband met diverse andere documenten worden in dit veld helaas soms weinig specifieke vermeldingen opgenomen. Ten aanzien van de opgave van het jaartal geldt een belangrijke opmerking. DIT MOET ALTIJD TUSSEN HAAKJES EN HET (IS HET EERSTE OPEN HAAKJE IN HET VELD!!! Bij bepaalde selecties op een gedeelte van de inhoud van dit veld werd dit haakje namelijk een rol toebedacht.

PLAATS:

Hierin staat een verwijzing naar de fysische plaats waar het stuk, het origineel, een kopie of een afdruk, kan gevonden worden.

AARD:

Hier wordt één van de volgende codes ingevoerd:

- | | |
|---|--|
| A | wijst op een Artikel in een klassiek tijdschrift |
| B | wijst op een Boek |
| C | wijst op een Citaat, een recensie van een boek |
| D | wijst op Diverse documenten |
| F | wijst op een artikel met een beperkte levensduur, bv iets in de lokale pers (Fleet street) |

Andere codes kunnen aangewend worden, maar hun behandeling is in dit programma niet voorzien. Evenmin is voorzien in een controle op de ingevoerde letters. Het programma is zo opgebouwd dat een andere code NIET tot een crash leidt. De codes spelen een rol in een controleprocedure die tussen komt bij het overschrijven van records van een tijdelijk bestand naar het standaardbestand op de harde schijf. De gebruiker dient te overwegen dat een andere code misschien niet op de door hem gewenste manier zal behandeld worden.

TREFWRD:

Dit veld is 120 tekens breed. Hierin worden trefwoorden opgenomen. Later kunnen records opgezocht worden met een stuk van een trefwoord of met een combinatie van trefwoorden.

MEMO:

De aanwezigheid van dit veld leidt er toe dat bij het creëren van het gegevensbestand automatisch een bestand met de extensie .dbt wordt aangemaakt. Hierin worden de memo's opgeslagen. Om velerlei redenen is dit een

ruimtevreter en een moeilijk te hanteren iets. Aanvankelijk was GEEN memoveld voorzien, maar het kan dat een brok informatie op een later tijdstip NIET meer toegankelijk zal zijn en dat hierin bepaalde interessante gegevens voorkomen. Ze kunnen in een memo worden bewaard. Het is ook mogelijk om een nieuw bestand te creëren en daar ruim van het memoveld gebruik te maken. Na het beëindigen van het voorgenomen werk kan dit bestand geschrapt worden en zo verdwijnt meteen ook het .dbt bestand.

Bestanden.

Het programma maakt van diverse bestanden gebruik. De namen worden als "defaultvalue" ingelezen bij het opstarten. Het programma controleert of het bestand voorhanden is. Indien dit niet het geval is, onder meer de allereerste keer dat het programma gebruikt wordt, gaat het programma zelf het standaardbestand en de indexbestanden aanmaken. Dit gebeurt met behulp van OPBOUW.dbf waarin de structuur van het aan te maken gegevensbestand opgenomen is.

De volgende bestanden zijn eigen aan het programma:

van het type dbf

LIT.dbf	HET gegevensbestand;
OPBOUW.dbf	een bestand waarin de structuur van vorige bestand opgenomen is;
WRKF_1.dbf	wordt als een soort indexbestand gebruikt. Dit 'indexbestand' wordt gemaakt op het ogenblik dat het vereist is. Het wordt vernietigd bij het verlaten van de module waarin het werd aangemaakt;
TANK.dbf	een gegevensbestand waarin steekkaarten tijdelijk opgeslagen worden. Dit gebeurt met het oog op een latere transfer naar het standaardbestand LIT.dbf;

van het type ndx

ANDX.ndx	het standaardindexbestand volgens de auteur;
DNDX.ndx	het standaardindexbestand volgens de drager;
PNDX.ndx	het standaardindexbestand volgens de plaats;
SNDX.ndx	het standaardindexbestand volgens de aard;
ATANK.ndx	het indexbestand volgens de auteur en gekoppeld aan TANK.dbf;
DTANK.ndx	het indexbestand volgens de drager en gekoppeld aan TANK.dbf;
PTANK.ndx	het indexbestand volgens de plaats en gekoppeld aan TANK.dbf;
STANK.dbf	het indexbestand volgens de aard en gekoppeld aan TANK.dbf;

van het type dbt

LIT.dbt
TANK.dbt

van het type qry

LIT.qry

Er dienen twee opmerkingen geformuleerd te worden:

1. het is mogelijk gegevensbestanden met een andere naam te creëren en te gebruiken. Het ligt voor de hand dat dan en de .dbf, de .dbt en de vier indexbestanden onder de aangepaste naam voorkomen;
2. er is een bijzonder bestand waarin de 'defaultwaarden' opgenomen zijn. De inhoud van dit bestand met de naam WAARDEN.mem, kan onder dBASE opgeroepen en gewijzigd worden. Zo kan het programma gemakkelijk aangepast worden aan bv de gebruikte printer of de gebezigde PC. De inhoud van dit bestand voor respectievelijk de desktop uitrusting, een SAMSUNG3000, en voor een geteste SANYO 16LT, een laptop met twee 720 kb drives, wordt hieronder weergegeven.

Inhoud "memory variabelen" voor de uitrusting SAMSUNG3000

```
KANTLLJN = 10
CONDENS = CHR(27)+CHR(15)           && gecondenseerd schrift
MERK = CHR(27)+CHR(126)+CHR(1)      && doorstreepte nul
NORMAAL = CHR(27)+CHR(64)           && initialiseren printer
DETPAPON = CHR(27)+CHR(57)          && koppelen papierdetector
DETPAPUT = CHR(27)+CHR(56)          && ontkoppelen detector
PLN = 72                             && papierlengte in lijnen
EINDE = .F.
VLAG = .T.                           && na printen .F.
DELAY_1 = 100                       && teller voor vertraging
SYSDRV = "C"                        && drive met systeemdiskette
TUS_F = "TANK.DBF"
ANDX = "D:\BIB\ALIT.NDX"
DNDX = "D:\BIB\DLIT.NDX"
PNDX = "D:\BIB\PLIT.NDX"
SNDX = "D:\BIB\SLIT.NDX"
NFILE = " "
NDX_1 = " "
NDX_2 = " "
NDX_3 = " "
NDX_4 = " "
STDF = "D:\BIB\LIT.DBF"
WRKF = "D:\BIB\WRKF_1.DBF"
OPBF = "D:\BIB\OPBOUW.DBF"
```

Inhoud "memory variabelen" voor de uitrusting SANYO 16LT

```
KANTLLJN = 10
CONDENS = CHR(27)+CHR(15)
MERK = CHR(27)+CHR(126)+CHR(1)
NORMAAL = CHR(27)+CHR(64)
DETPAPON = CHR(27)+CHR(57)
DETPAPUT = CHR(27)+CHR(56)
PLN = 72
EINDE = .F.
VLAG = .T.
DELAY_1 = 100
SYSDRV = "A"
```

```
TUS_F = "TANK.DBF"  
ANDX = "ALIT.NDX"  
DNDX = "DLIT.NDX"  
PNDX = "PLIT.NDX"  
SNDX = "SLIT.NDX"  
NFILE = " "  
NDX_1 = " "  
NDX_2 = " "  
NDX_3 = " "  
NDX_4 = " "  
STDF = "LIT.DBF"  
WRKF = "WRKF_1.DBF"  
OPBF = "OPBOUW.DBF"
```

PROGRAMMAMOGELIJKHEDEN.

De programmagemogelijkheden worden toegelicht aan de hand van de interactieve communicatie tussen het programma en de gebruiker. In het geheel kunnen zo'n zeven delen onderscheiden worden, met name:

- het opstarten
- het toevoegen van steekkaarten
- het bewerken van de inhoud van steekkaarten (redigeren)
- het schrappen van steekkaarten
- het kopiëren van steekkaarten
- andere opdrachten (toolbox)
- het sluiten

Het opstarten.

Zowel het starten van de PC, het laden van MS-DOS en van dBASEIII+ worden hier niet behandeld. Hiervoor wordt verwezen naar de diverse handleidingen. Na het voltooien van deze voorbereidende werkzaamheden, wordt het eigenlijke programma opgeroepen via het puntkommando DO PROG. Het venster aangeduid als SCHRM01 verschijnt.

SCHRM01:

Programma voor beheer referenties
Versie 1.31
Oostende *** november 1990
W. Deschacht

Zorg voor een voldoende voorraad
geformatteerde diskettes voor het
uitvoeren van een backup

<CR> aantikken.....

SCHRM01

Men wordt aangeraden te zorgen voor floppies om probleemloos een kopie van het gegevensbestand aan te kunnen maken bij het verlaten van het programma. Mocht men nu NIET over geformatteerde diskettes beschikken, dan is er niets aan de hand. Het formatteren is mogelijk onder de controle van het

programma zelf. De ENTER toets (Carriage Return) aanslaan doet het programma onmiddellijk verder lopen. Wordt de CR toets NIET aangeraakt, dan verdwijnt het openingsscherm na een tiental seconden en de gebruiker wordt uitgenodigd EEN van de in SCHRM02 geboden opties te kiezen.

SCHRM02:

Werken met...

De beschikbare opties...

[0] het programma afbreken
[1] standaardbestand
[2] ander bestand

Je keuze aub? <0..2> *

SCHRM02

SCHRM03:

Database Files	# Records	Last Update	Size
LIT.DBF	3461	10/01/90	1356970
OPBOUW.DBF	7	08/06/90	288
WRKF_1.DBF	0	10/01/90	512
TANK.DBF	0	10/01/90	258

1358028 bytes in 4 files
23838720 bytes remaining on drive

Dit zijn de gebruikte bestandsnamen

SCHRM03

Het programma wordt verlaten bij het intoetsen van een NUL. Valt de keuze op EEN, dan worden de standaardnamen voor het gegevensbestand en de indexbestanden aangegeven aan het programma en dit laatste belandt bij het hoofdmenu.

SCHRM04:

Dit zijn voorbehouden bestandsnamen...

D:\BIB\LIT.DBF
TANK.DBF
D:\BIB\WRKF_1.DBF
D:\BIB\OPBOUW.DBF

SCHRM04

Bij het aantikken van de toets <2> wordt eerst een scherm getoond met de voorhanden zijnde gegevensbestanden, gevolgd, na een korte periode, door een lijst met de voorbehouden namen. Ook dit scherm verdwijnt na een korte tijd

en dan verschijnt de vraag naar de naam van het bestand. Voor de gebruikte PC wordt een en ander in de afbeeldingen SCHRM03 tot en met SCHRM05 voorgesteld.

Het programma wordt beëindigd door de ESC-toets in SCHRM05 aan te tikken. Het intypen van de naam gebeurt in een balk in revers video. Na het aantikken van de CR-toets maakt het programma de namen van de indexbestanden aan. Vervolgens wordt gecontroleerd of het bestand bestaat. Is dit niet het geval dan worden de bestanden aangemaakt. Uiteindelijk worden de namen overgemaakt aan het programma en verschijnt het hoofdmenu. Dit wordt in SCHRM06 weergegeven.

SCHRM05:

De volledige pathnaam bestand.DBF... (STOPPEN: ESC)

SCHRM05

SCHRM06:

H O O F D - M E N U

De beschikbare opties:

- [0] het programma verlaten
- [1] steekkaart(en) toevoegen
- [2] steekkaart(en) redigeren
- [3] steekkaart(en) wissen
- [4] steekkaart(en) kopiëren
- [5] andere opdracht

Je keuze aub? <0..5> *

SCHRM06

Een optie wordt gekozen door het cijfer in te toetsen. Het programma controleert de geldigheid van de keuze. Bij een niet-geldige keuze kunnen twee mogelijkheden zich voordoen:

- a) er werd een niet-numeriek gegeven ingetoetst:
het programma wacht op een nieuwe invoer;
- b) er werd een numeriek gegeven met een te hoge waarde ingetoetst:
in de rechterbovenhoek van het scherm verschijnt het geldige interval en het verzoek de spatiebalk aan te tikken, waarna het programma wacht op een nieuwe invoer.

Op de ingebouwde controle(s) bij het maken van een keuze wordt niet meer expliciet teruggekomen.

Steekkaarten toevoegen.

SCHRM07:

Kiest men in het hoofdmenu voor de optie 1, dan verschijnt SCHRM07. De eerste mogelijkheid voert naar het hoofdmenu (SCHRM06). Optie 1 maakt het mogelijk dat aan het middels SCHRM05 gekozen bestand records worden toegevoegd. Deze optie geeft aanleiding tot het verschijnen van SCHRM09. Bij optie 2 worden de toe te voegen records in het bestand TANK.DBF geschreven (conf. inf.). Indien deze techniek gevolgd wordt, kunnen later de steekkaarten naar het gegevensbestand overgebracht worden met behulp van optie 3 (SCHRM14). De keuze van optie 4 leidt tot de vernietiging van het tijdelijke bestand.

STEEKKAART(EN) TOEVOEGEN

De beschikbare opties:

- [0] de module verlaten
- [1] toevoegen aan permanent bestand
- [2] toevoegen aan tijdelijk bestand
- [3] transfer van tijd. naar perm. bestand
- [4] schrappen tijdelijk bestand

Je keuze aub? <0..4> *

SCHRM07

SCHRM08:

Wordt in SCHRM07 optie 2 gekozen, dan verschijnt SCHRM08. Veel toelichting is hierbij niet vereist. De nul-optie brengt de gebruiker terug naar het SCHRM07. A, B, C en D verwijzen uiteraard naar de drive waarin de disk(ette) met het hulpbestand voorhanden is. Nogmaals zij er op gewezen dat het hulpbestand en de daarbij horende indexen zonodig worden gecreëerd. Een geldige, van NUL verschillende keuze doet SCHRM09 verschijnen. Dit is ook het geval indien optie 1 in SCHRM07 werd gekozen.

STEEKKAART(EN) TOEVOEGEN AAN HULPBESTAND

Localisatie van het hulpbestand:

- [0] de module verlaten
- [1] hulpbestand op A:
- [2] hulpbestand op B:
- [3] hulpbestand op C:
- [4] hulpbestand op D:

Je keuze aub? <0..4> *

SCHRM08

SCHRM09:

De nul-optie brengt de gebruiker terug in SCHRM08. Indien optie 1 weerhouden wordt, dan controleert het programma het bestand op de aanwezigheid van een steekkaart met de gebruikte naam in het veld AUTEUR. Een ana-

loog verloop kennen de opties 2 en 3 maar dan gebeurt de controle op de inhoud van het veld TITEL of DRAGER. De optie 2 is een tijdverslindende keuze vermits dit veld niet geïndexeerd is. Deze mogelijkheid is belangrijk voor de controle bij de invoer van een boek of van een recensie van een boek. De optie 4 is erg bruikbaar voor het opnemen van gegevens uit een pas ontvangen nr van een tijdschrift. Deze keuze voert meteen naar het invoerscherm (SCHRM10), een andere keuze leidt tot SCHRM11.

KONTROLE BIJ DE INVOER

De beschikbare opties:

- [0] de module verlaten
- [1] middels de auteur
- [2] middels de titel
- [3] middels de drager
- [4] geen

Je keuze aub? <0..4>

*

SCHRM09

SCHRM10:

GEGEVENS INTOETSEN

Auteur :

Titel :

Drager :

Plaats :

Trefwoord :

cursor beweging	wissen	einde
↑	HOME	letter: Del
← →	END	woord : ^T
↓		veld : ^Y
		Opslaan: ^END
		^W
		Afbreken: ESC

SCHRM10

De invoer gebeurt in een balk waarvan de lengte NIET altijd gelijk is aan de lengte van het veld. Voor de velden titel, drager en trefwoord wordt een vensterbreedte van 60 tekens gebruikt. Bij de invoer van een groter aantal tekens schuift de tekst door tot het aantal beschikbare plaatsen opgebruikt is. Het aanslaan van de toetsencombinatie <Ctrl><END> (^END) of <Ctrl><W> (^W) voert naar het invoerscherm voor de MEMO. Het dBASEIII+ hulp-

scherm geeft aan hoe dit verlaten wordt na het al dan niet noteren van gegevens. De Esc-toets aanslaan dwingt het programma om het invoerscherm te verlaten en de reeds ingevulde gegevens van het betrokken record gaan daarbij verloren.

SCHRM11:

In de veronderstelling dat de controle via de auteur gebeurt, zet het programma SCHRM11 op de buis. Wordt de ESC-toets aangetikt, dan wordt dit programmaonderdeel verlaten en men komt in het aanroepend menu (SCHRM09). Voert men een naam in, dan wordt aan het scherm een vraag toegevoegd. Dit wordt in SCHRM12 getoond.

Druk F10 om vorige invoer te herhalen

Auteur... (exakt *** STOPPEN: ESC)

SCHRM11

SCHRM12:

Druk F10 om vorige invoer te herhalen

Auteur... (exakt *** STOPPEN: ESC)

PRIMROSE S.

Is dit gegeven OK? <j/n>

SCHRM12

Een negatief antwoord op de gestelde vraag plaatst de cursor op de P uit het auteursveld en de nodige verbeteringen kunnen worden aangebracht. Is de zaak OK dan gaat het programma over tot de controle. Komt deze auteur nog niet in het bestand voor, dan verschijnt het invoerscherm (SCHRM10) anders het leesscherm (SCHRM13). In tegenstelling tot het getoonde SCHRM10 staat dan het testwoord in het gecontroleerde veld. Werd alreeds een record ingevuld, dan zal de inhoud van het veld waarop de controle gebeurt opgeroepen worden door de functietoets F10 te drukken. Wanneer uit een of meerdere nummers van een gegeven tijdschrift artikels opgenomen worden, zal het aantikken van de F10 toets de testwaarde van het vorige record in het testveld plaatsen, in dit geval allicht het veld drager, en een kleine aanpassing, bv de pagina, is voldoende om de gehele procedure opnieuw te doorlopen.

SCHRM13

De vermelding in SCHRM13 leert dat het nutteloos is om de keuze op A (= voorafgaande record) te laten vallen. Verder wordt duidelijk gemaakt dat hier een geïndexeerd bestand gebezigd wordt. Ware dit het eerste record in het fysische bestand geweest dan luidde de melding: "Eerste record in het fysische bestand". Het laat zich makkelijk raden wat te lezen zou zijn indien dit record het laatste in een logisch dan wel een fysisch bestand ware. De A en P toets laten de gebruiker toe om te grasduinen door de steekkaarten die aan het criterium beantwoorden. Wordt de E aangetikt, dan keert het programma

loog verloop kennen de opties 2 en 3 maar dan gebeurt de controle op de inhoud van het veld TITEL of DRAGER. De optie 2 is een tijdverslindende keuze vermits dit veld niet geïndexeerd is. Deze mogelijkheid is belangrijk voor de controle bij de invoer van een boek of van een recensie van een boek. De optie 4 is erg bruikbaar voor het opnemen van gegevens uit een pas ontvangen nr van een tijdschrift. Deze keuze voert meteen naar het invoerscherm (SCHRM10), een andere keuze leidt tot SCHRM11.

KONTROLE BLJ DE INVOER

De beschikbare opties:

- [0] de module verlaten
- [1] middels de auteur
- [2] middels de titel
- [3] middels de drager
- [4] geen

Je keuze aub? <0..4> *

SCHRM09

SCHRM10:

GEGEVENS INTOETSEN

Auteur :
 Titel :
 Drager :
 Plaats :
 Trefwoord :

cursor beweging		wissen	einde
↑	HOME	letter: Del	Opslaan: ^END
← →	END	woord : ^T	^W
↓		veld : ^Y	Afbreken: ESC

SCHRM10

De invoer gebeurt in een balk waarvan de lengte NIET altijd gelijk is aan de lengte van het veld. Voor de velden titel, drager en trefwoord wordt een vensterbreedte van 60 tekens gebruikt. Bij de invoer van een groter aantal tekens schuift de tekst door tot het aantal beschikbare plaatsen opgebruikt is. Het aanslaan van de toetsencombinatie <Ctrl><END> (^END) of <Ctrl><W> (^W) voert naar het invoerscherm voor de MEMO. Het dBASEIII+ hulp-

scherm geeft aan hoe dit verlaten wordt na het al dan niet noteren van gegevens. De Esc-toets aanslaan dwingt het programma om het invoerscherm te verlaten en de reeds ingevulde gegevens van het betrokken record gaan daarbij verloren.

SCHRM11:

In de veronderstelling dat de controle via de auteur gebeurt, zet het programma SCHRM11 op de buis. Wordt de ESC-toets aangetikt, dan wordt dit programmaonderdeel verlaten en men komt in het aanroepend menu (SCHRM09). Voert men een naam in, dan wordt aan het scherm een vraag toegevoegd. Dit wordt in SCHRM12 getoond.

Druk F10 om vorige invoer te herhalen

Auteur... (exakt *** STOPPEN: ESC)

SCHRM11

SCHRM12:

Druk F10 om vorige invoer te herhalen

Auteur... (exakt *** STOPPEN: ESC)

PRIMROSE S.

Is dit gegeven OK? <j/n>

SCHRM12

Een negatief antwoord op de gestelde vraag plaatst de cursor op de P uit het auteursveld en de nodige verbeteringen kunnen worden aangebracht. Is de zaak OK dan gaat het programma over tot de controle. Komt deze auteur nog niet in het bestand voor, dan verschijnt het invoerscherm (SCHRM10) anders het leesscherm (SCHRM13). In tegenstelling tot het getoonde SCHRM10 staat dan het testwoord in het gecontroleerde veld. Werd alreeds een record ingevuld, dan zal de inhoud van het veld waarop de controle gebeurt opgeroepen worden door de functietoets F10 te drukken. Wanneer uit een of meerdere nummers van een gegeven tijdschrift artikels opgenomen worden, zal het aantikken van de F10 toets de testwaarde van het vorige record in het testveld plaatsen, in dit geval allicht het veld drager, en een kleine aanpassing, bv de pagina, is voldoende om de gehele procedure opnieuw te doorlopen.

SCHRM13

De vermelding in SCHRM13 leert dat het nutteloos is om de keuze op A (= voorafgaande record) te laten vallen. Verder wordt duidelijk gemaakt dat hier een geïndexeerd bestand gebezigd wordt. Ware dit het eerste record in het fysische bestand geweest dan luidde de melding: "Eerste record in het fysische bestand". Het laat zich makkelijk raden wat te lezen zou zijn indien dit record het laatste in een logisch dan wel een fysisch bestand ware. De A en P toets laten de gebruiker toe om te grasduinen door de steekkaarten die aan het criterium beantwoorden. Wordt de E aangetikt, dan keert het programma

terug naar SCHRM11, wordt N aangetikt dan verschijnt het inschrijfscherm en de velden kunnen ingevuld worden.

Dit is het eerste record

Gevonden steekkaart

Auteur : PRIMROSE S.
Titel : INTRODUCTION TO MODERN VIROLOGY

Drager : BLACKWELL SCIENTIFIC PUBLICATIONS, OXFORD (1978) ISBN
0-632-09230-0

Plaats : PI Aard: B

Trefwoord : VIROLOGIE LEERBOEK MICROBIOLOGIE

A(nte, P(ost, N(oteren, E(xit

Je keuze aub? <A/P/N/E>

SCHRM13

Werd in SCHRM09 de toets <4> gedrukt, dan verloopt het invoeren zonder controle. Telkenmale een steekkaart ingevuld en weggeschreven is komt op het scherm de vraag of er nog een steekkaart dient ingevuld te worden. In geval van een bevestigend antwoord verschijnt het invoerscherm (SCHRM10) en anders belandt het programma bij SCHRM09. Bij de niet gecontroleerde invoer speelt de F10 toets niet mee.

SCHRM14:

Wordt in SCHRM07 de optie 3 gekozen, dan verschijnt SCHRM14. De vermelding "D:\BIB\LIT.DBF" komt slechts op het scherm wanneer dit inderdaad het standaardbestand is. De opgave van een andere "default" in het WAARDEN.mem of de keuze van een ander bestand in SCHRM05 toont de eventueel aangepaste benaming.

TRANSFER STEEKKAART(EN) NAAR PERM. BESTAND

De beschikbare opties:

- [0] de module verlaten
- [1] transfer van A: -> D:\BIB\LIT.DBF
- [2] transfer van B: -> D:\BIB\LIT.DBF
- [3] transfer van C: -> D:\BIB\LIT.DBF
- [4] transfer van D: -> D:\BIB\LIT.DBF

Je keuze aub? <0..4> *

SCHRM14

De keuze 0 voert terug naar SCHRM07. Een andere, geldige keuze leidt tot de uitvoering van de betreffende opdracht. In dit geval wordt gewezen op de noodzaak om de printer in te schakelen. Dit gebeurt via het scherm SCHRM15. De printer is nodig omdat het programma het in SCHRM05 weerhouden bestand controleert op het voorhanden zijn van een steekkaart. Daartoe wordt een combinatie van twee criteria gebruikt. Beslist het programma tot het gelijk zijn van de steekkaart aan een reeds voorhanden zijnde record dan worden de gegevens afgedrukt. De gebruiker kan later een controle uitvoeren en alsnog beslissen de gegevens toch op te moeten nemen. Wanneer alle records uit het tijdelijke bestand behandeld zijn, keert het programma terug naar SCHRM02.

SCHRM15:

Zorg voor voldoende kettingpapier...
Zet printkop te paard op scheurlijn...
Schakel stroom in...
Zet printer ON LINE...

<H> = uitvoeren hoofdmenu
andere toets = verder met uitvoering...

SCHRM15

SCHRM16:

De standaardlengte bedraagt 66 of 72 lijnen

De aktuele lengte belooft 72 lijnen

Indien OK druk <O>
anders <V>

SCHRM16

SCHRM17:

TIJDELIJK BESTAND SCHRAPPEN

Localisatie van het hulpbestand:

[0] de module verlaten
[1] hulpbestand op A:
[2] hulpbestand op B:
[3] hulpbestand op C:
[4] hulpbestand op D:

Je keuze aub? <0..4> *

SCHRM17

De H-toets voert het scherm SCHRM02 ten tonele! Een andere toets levert SCHRM16 op. Wordt de O-toets ingedrukt dan gebruikt het programma voor de printer de aktuele papierlengte (hier 72), aantikken van de V-toets stelt de

papierlengte in op 66 lijnen. Het aantikken van een van beide toetsen brengt de transferprocedure op gang.

Wordt in SCHRM07 voor de optie 4 gekozen, dan verschijnt SCHRM17. Een 0 brengt het SCHRM07. Een andere geldige keuze leidt tot de vernietiging van het tijdelijke bestand en de daarmee verbonden .ndx en .dbt bestanden.

Steekkaarten redigeren.

De behandeling van de inhoud van bestaande steekkaarten wordt ingezet met de optie 2 uit SCHRM06. Deze keuze voert tot SCHRM18.

SCHRM18:

STEEKKAART(EN) REDIGEREN

De beschikbare mogelijkheden:

- | | |
|-----|---------------------------------------|
| [0] | de module verlaten |
| [1] | zoeken via auteur |
| [2] | zoeken via drager |
| [3] | zoeken via plaats |
| [4] | zoeken via EEN trefwoord |
| [5] | zoeken via EEN recordnummer |
| [6] | zoeken via een interval aan recordnrs |
| [7] | zoeken via aard |
| [8] | anders zoeken |

Je keuze aub? <0..8> *

SCHRM18

De nul brengt het programma terug naar het hoofdmenu (SCHRM06), de opties 1, 2, 3, 4, 5 en 7 vragen de naam van de auteur, de drager enz. Hierbij moet aangestipt worden dat ook het begin van een naam volstaat om het zoeken toe te laten. In de hierboven geciteerde steekkaart PRIMROSE S. zou het aangeven van PRIMROSE voldoende zijn. Overigens krijgt men na het aantikken van de CR-toets in SCHRM19 nog de gelegenheid tot verbeteren. SCHRM19 wordt met een vraag aangevuld zoals weergegeven in SCHRM20. Een bevestigend antwoord brengt de zoekprocedure op gang, een ontkenning plaatst de cursor op de eerste letter in het invoerveld van SCHRM19. Een en ander eindigt met het vinden of niet vinden van de gevraagde steekkaart. Indien bv PRIMROSE niet als auteur in het bestand voorhanden is, verschijnt daaromtrent een mededeling op het scherm. Dit wordt voorgesteld in SCHRM21.

SCHRM19:

F10 = vorige invoer

auteur... (STOPPEN: ESC)

PRIM

SCHRM19

Een bijzondere vorm van niet vinden doet zich voor bij het gebruik van EEN recordnummer. Misschien weet de gebruiker niet precies hoeveel steekkaarten in zijn bestand opgeslagen waren en vraagt hij een niet bestaand record. Het programma reageert hierop met de mededeling uit SCHRM22. Wordt het gegeven wel gevonden, dan verschijnt de inhoud van de steekkaart op het scherm zoals voorgesteld door SCHRM26. In verband met het aanwenden van de inhoud van het veld DRAGER als zoekcriterium kan vermeld worden dat een gedeeltelijke invoer enkele interessante selectiemogelijkheden inhoudt. Dit wordt door de volgende voorbeelden toegelicht:

SCI. AMER.

alle artikels in SCIENTIFIC AMERICAN verschenen en opgenomen in het bestand;

SCI. AMER. (1977)

voor zover in het gegevensbestand opgenomen alle artikels in de loop van het jaar 1977 verschenen in SCIENTIFIC AMERICAN;

SCI. AMER. (1977)236

alle artikels uit volume 236;

SCI. AMER. (1977)236(1)

alle artikels uit het nr 1 van de betrokken jaargang en volume;

SCI. AMER. (1977)236(1)28

het artikel dat aanvangt op blz 28 van het geciteerde nr, volume en jaargang.

SCHRM20:

F10 = vorige invoer

auteur... (STOPPEN: ESC)

PRIMROSE

is dit gegeven OK? <J/N>

SCHRM20

De betekenis van het drukken op de ESC-toets voor de SCHRM19 en 20 zal wel duidelijk zijn. Het programma keert terug naar SCHRM18. Het aantikken van een willekeurige toets in SCHRM21 of SCHRM22 brengt het programma naar

SCHRM21:

Het gegeven

PRIMROSE

werd NIET gevonden

Druk op een willekeurige toets...

SCHRM21

SCHRM19. Alvorens het leesscherf te bekijken resten nog twee duidelijk afwijkende zoekprocedures te behandelen. De afbeeldingen uit SCHRM23 en SCHRM24 illustreren de gang van zaken voor het invoeren van een interval aan recordnrs. Een 'j' of 'J' antwoorden op de in SCHRM24 gestelde vraag leidt

tot de mogelijkheid om het laatste te raadplegen record op te geven. Een tweede bevestigend antwoord doet in onderhavig voorbeeld het record met volg-nr 115 op het scherm verschijnen.

SCHRM22:

Het hoogste recordnr is 3462

Druk een willekeurige toets...

SCHRM22

SCHRM23:

laatste recordnr in bestand ... 1752

beginnen bij recordnr... 0

SCHRM23

SCHRM24:

laatste recordnr in bestand... 1752

beginnen bij recordnr... 115

Is dit gegeven OK? <j/n> j

eindigen met recordnr... 134

Is dit gegeven OK? <j/n>

SCHRM24

SCHRM25:

Bij het aantikken van de optie 8 in SCHRM18 verschijnt de mededeling van SCHRM25.

Zo direct verschijnt een hulpscherm
waarmede de selectievoorwaarden kunnen
vastgelegd worden.

Onthou:

lijn wissen: lijnnr kiezen <^U>
lijn invoegen: lijnnr kiezen <^N>
gebruiken: SAVE

Willekeurige toets drukken...

SCHRM25

Het aantoezen van een willekeurige toets doet het queryscherm uit dBASEIII+ verschijnen. Op het gebruik hiervan wordt niet nader ingegaan. Dit gebeurt

uitvoerig in de dBASEIII+ handleiding. De hier verstrekte gegevens en de op het scherm geprojecteerde onderrichtingen zijn voldoende om zonder grote problemen van deze krachtige, maar trage zoekprocedure van dBASEIII+ gebruik te maken. Het zoeken start door de optie SAVE te kiezen. Na het beëindigen van een zoekopdracht verschijnt op het scherm de vraag of er nog een selectie dient te gebeuren. Een bevestigend antwoord brengt het programma naar SCHRM25, een ontkenning naar SCHRM18.

SCHRM26:

dit is het eerste record

Gevonden steekkaart

Auteur : HOVART P.
Titel : ENERGIE EN ZEEVISSERIJ

Drager : AGRICONTACT (1982)132(VA)1

Plaats : PI-9/00
Trefwoord : ENERGIE ZEE

Aard : A

A(nte P(ost W(ijzig S(chrap O(pvissen M(emo E(xit *
auteur... HOVART

SCHRM26

Op dit scherm komen een aantal mededelingen voor. Hier is het niet nodig de A aan te toetsen, vermits het eerste record getoond wordt. De P mag wel. Moest het getoonde record meteen ook het laatste record zijn, dan zou de mededeling luiden: "dit is een enig record". Wordt P aangetikt, dan verschijnt het volgende record dat beantwoordt aan het criterium:

auteur... HOVART

De mededeling betreffende de rangorde van het record verdwijnt dan. Zo kunnen alle records konform aan het criterium doorbladerd en zo nodig aangepast worden. Komt het laatste record op dan verschijnt de mededeling: "dit is het laatste record". Mocht een voor schrappen gemerkt record gevonden worden, dan verschijnt op een andere regel de boodschap:

logisch geschrapt

Kiest men in zo'n geval de optie O dan wordt deze schrapping ongedaan gemaakt. Via de S kan een steekkaart logisch geschrapt worden en de W aantikken laat toe de inhoud van de velden aan te passen. In zo'n geval verschijnt SCHRM27. Alvorens dit scherm te behandelen weze nog aangestipt dat bij het zoeken via EEN recordnr de mogelijkheden uit het SCHRM26 beperkt zijn tot:

W(ijzigen S(chrappen O(pvissen M(emo E(xit

SCHRM27:

Na het aanbrengen van de wijzigingen kan men het redigeerscherm verlaten. De behandelde gegevens worden opgeslagen en SCHRM26, gevuld met de inhoud van het gewijzigde record, wordt getoond. Links boven staat de melding: "gewijzigd".

Auteur HOVART P.

Titel ENERGIE EN ZEEVISSERIJ

Drager AGRICONTACT (1982) 132 (VA) 1

Plaats PI-9/00

Aard : A

Trefwoord ENERGIE ZEE VISSERIJ

Cursor beweging	Wissen	Einde
↑	HOME	letter : Del
← →	END	woord : ^T
↓		veld : ^Y
		Opslaan: ^END
		^W
		Afbreken: ESC

SCHRM27

Steekkaarten wissen.

De optie 3 in SCHRM06 leidt tot SCHRM28. Wordt hierin de 0 optie gekozen, dan brengt dit het programma terug tot het hoofdmenu, SCHRM06, de optie 1 laat toe bepaalde records te merken zodat ze later fysisch kunnen gewist worden. Het zoeken gebeurt aan de hand van criteria die op dezelfde wijze vastgelegd worden als de criteria bij het redigeren (SCHRM18). Alleen de titel is verschillend en luidt nu: STEEKKAART(EN) WISSEN. In verband met het verdere verloop van de operaties zij verwezen naar de figuren SCHRM19 tot en met SCHRM27 en de daarbij gegeven toelichtingen.

SCHRM28:

STEEKKAART(EN) WISSEN

De beschikbare opties:

- [0] de module verlaten
- [1] steekkaart(en) logisch wissen
- [2] controle gewiste steekkaart(en)
- [3] steekkaart(en) fysisch wissen

Je keuze aub? <0..3>

*

SCHRM28

Indien de optie 2 weerhouden werd, loopt het programma het standaard-, of het in SCHRM05 aangeduide, bestand af en toont ieder logisch geschrapte record. De operator krijgt de gelegenheid een schrapping ongedaan te maken. De opties uit het leesscherm (SCHRM26) zijn beperkt tot:

A(nte P(ost W(is O(pvissen E(xit

of in het geval gezocht wordt via EEN recordnr:

W(is O(pvissen E(xit

SCHRM29:

de gemerkte records worden verwijderd

dit kan enige tijd in beslag nemen

3461 records copied
Rebuilding index - D:\BIB\ALIT.NDX
100 % indexed 3461 Records indexed
Rebuilding index - D:\BIB\DLIT.NDX
100 % indexed 3461 Records indexed
Rebuilding index - D:\BIB\PLIT.NDX
100 % indexed 3461 Records indexed
Rebuilding index - D:\BIB\SLIT.NDX
36 % indexed

SCHRM29

Werd in SCHRM28 de optie 3 gekozen dan wordt het scherm gewist, er verschijnt een mededeling die naderhand aangevuld wordt met informatie over het herstel van de indexbestanden. SCHRM29 illustreert dit alles.

Steekkaarten kopiëren.

SCHRM30:

Wanneer in het SCHRM06 geopteerd wordt voor de mogelijkheid 4 dan verschijnt SCHRM30.

Wordt hierin de 0 optie weerhouden, dan komt het programma met het hoofdmenu uit SCHRM06 voor de proppen. Iedere andere keuze resulteert in een scherm waarop een aantal keuzen vermeld worden. Deze staan in verband met de criteria die bij het kopiëren gebezigd worden. In feite gebruikt het programma de mogelijkheden uit SCHRM18. Er werd voor iedere geval een aangepaste titel voorzien. De verschillende opschriften luiden:

RECORDS NAAR SCHERM SCHRIJVEN
RECORDS AFDRUKKEN
RECORDS KOPIEREN OP SCHIJF IN A:
RECORDS KOPIEREN OP SCHIJF IN B:
RECORDS KOPIEREN OP SCHIJF IN C:
RECORDS KOPIEREN OP SCHIJF IN D:

Wanneer records afgedrukt moeten worden gebeurt dit zonder de memo. Indien de printer tot op dit ogenblik niet gebruikt werd verschijnt SCHRM15.

Na de keuze van het criterium wordt hiervan de inhoud gevraagd met behulp van schermen schermen vergelijkbaar aan deze uit de illustraties SCHRM19 tot en met SCHRM25. De daar gegeven toelichting geldt ook hier. Vervolgens wordt met behulp van schermen analoog aan deze uit de figuren SCHRM19 en SCHRM20 de naam van het doelbestand gevraagd. Bestaat het bestand waar de records naartoe moeten gekopieerd worden, dan verschijnt SCHRM31, bestaat het niet, dan wordt het gecreeërd.

STEEKKAART(EN) KOPIEREN

De beschikbare opties:

- [0] de module verlaten
- [1] kopie naar scherm
- [2] kopie naar printer
- [3] kopie naar floppy in A:
- [4] kopie naar floppy in B:
- [5] kopie op HDISK (C:)
- [6] kopie op HDISK (D:)

Je keuze aub? <0..6> *

SCHRM30

SCHRM31:

de naamdoelbestand bestaat

O(verschrijven S(toppen T(oevoegen) *)

SCHRM31

Wordt de O-optie gekozen, dan zal het bestand gebruikt worden, maar de records zullen door de nieuwe gegevens overschreven worden. Bij het intikken van S keert het programma terug naar het aanroepend menu, dit is een scherm vergelijkbaar aan SCHRM18 maar met de overeenstemmende titel. De T-optie tenslotte laat toe de gekopieerde records aan het bestaande bestand toe te voegen. Bij het kopiëren van een bestand of een gedeelte van het bestand wordt automatisch onder de overeenstemmende benaming een .dbt bestand aangemaakt. Er worden nochtans GEEN indexen opgesteld, zodat een raadplegen van dit bestand via dit programma slechts mogelijk is nadat de gepaste indexbestanden met behulp van de dBASEIII+ mogelijkheden werden aangemaakt.

Andere opdracht.

Wordt in SCHRM06 de optie 5 gekozen, dan belandt het programma in SCHRM32. De 0-optie brengt het programma terug in SCHRM06. De optie 1 geeft aanleiding tot het verschijnen van SCHRM33.

De nul-optie voert het programma meteen naar SCHRM06 terug. Wordt 1 ingetoetst dan vraagt het programma welke drive de floppy bevat waarop de backup zal gebeuren. Dit gebeurt door middel van een invoerscherm analoog aan SCHRM19 en SCHRM20. Zoals steeds kan men middels de ESC-toets de lopende procedure verlaten en dan komt het programma terecht in SCHRM32. Voor

SCHRM32:

TOOLBOX

De beschikbare opties:

- [0] de module verlaten
- [1] backup uitvoeren
- [2] lijst periodieken afdrukken
- [3] formatteren van floppies

Je keuze aub? <0..3> *

SCHRM32

een andere keuze controleert het programma vanaf welke drive het besturings-systeem werd opgestart. Voor de hier gebruikte SAMSUNG3000 was dit de C:, maar bij de testen met de SANYO 16LT was dit de A: drive. In het eerste geval wordt meteen de BACKUP opdracht van MS-DOS geactiveerd waardoor het .dbf en het .dbt bestand behandeld worden. In het tweede geval deelt het programma mede dat het nodig is de MS-DOS diskette in A: te plaatsen. Het programma kan verder gezet worden door een willekeurige toets in te drukken. Een en ander wordt voorgesteld in SCHRM34.

SCHRM33:

Backup-Menu

De beschikbare opties:

- [0] Terug naar hoofdmenu
- [1] Backup van HD naar floppy
- [2] Restore van floppy naar HD

Je keuze aub? <0..2> *

SCHRM33

SCHRM34:

plaats MS-DOS diskette in A:

druk willekeurige toets...

SCHRM34

Wordt de keuze 2 ingevoerd dan vraagt het programma de drive waarin de floppy zich bevindt van waarop het herstel van het standaardbestand zal gebeuren, de zogenaamde brondrive. Een en ander gebeurt via SCHRM19 en SCHRM20. Ook hier kan de lopende module verlaten worden door de ESC-toets aan te tikken. Dit voert het programma naar het TOOLBOX-menu (SCHRM32). In het andere geval wordt op de hierboven beschreven manier een controle uitgevoerd in verband met de drive van waaruit het MS-DOS programma opgestart werd. Is dit de C: dan wordt de RESTORE opdracht uit MS-DOS geactiveerd en de betrokken .dbf en .dbt bestanden worden naar de harddisk gekopieerd. Is dit de A: drive dan verschijnt de boodschap uit SCHRM34. Aantikken van de willekeurige

toets zet dan de RESTORE opdracht van MS-DOS aan het werk. Nadat de bestanden in de D:\BIB subdirectory gekopieerd zijn, worden de indexbestanden geopend en hersteld. In dit verband kan nog een opmerking worden geformuleerd. Een RESTORE is normaliter slechts nodig nadat, om welke reden dan ook, het originele bestand beschadigd werd. Dit kan aanleiding geven tot het starten bij het begin van het programma, zodat SCHRM02 noopt tot een eerste keuze, en dit wordt dan toets <2>. Als gevolg hiervan verschijnt dan uiteindelijk SCHRM05 en dient, in de hierboven geschetste omstandigheden, D:\BIB\LIT.DBF als naam meegegeven te worden.

Zo tot op dat ogenblik nog geen gebruik van de printer gemaakt werd, zal het intoetsen van 2 in SCHRM32, leiden tot het verzoek de nodige maatregelen te treffen om de printer te kunnen gebruiken. Dit gebeurt middels SCHRM15 en SCHRM16. Het programma drukt daarna een lijst af waarin van ieder tijdschrift (aard = A of F) één voorbeeld voorhanden is van de wijze waarop het veld DRAGER ingevuld werd. Na afloop komt het programma in SCHRM02 terecht.

SCHRM35:

```

                        Formatteren van floppies
                        *****

De beschikbare opties:

[0]      naar hoofdmenu
[1]      in de A: drive
[2]      in de B: drive

Je keuze aub?      <0..2>      *
```

SCHRM35

Viel in SCHRM32 de keuze op optie 3 dan verschijnt SCHRM35. Een keuze 0 voert meteen tot SCHRM02. Een andere keuze doet het programma een controle uitvoeren in verband met de drive van waaruit de PC opgestart werd. Is dit de C: drive dan wordt het kommando FORMAT uitgevoerd op de gekozen drive. Is dit de A: drive dan wordt de operator verzocht een gepaste MS-DOS diskette in de A: drive te plaatsen en door een willekeurige toets aan te tikken wordt het FORMAT kommando geactiveerd. De gebruiker moet dan de aanwijzingen volgen die dit kommando op het scherm projecteert.

Sluiten.

Indien in SCHRM02 of in SCHRM06 geopteerd wordt voor de keuze 0 dan zorgt het programma voor een ordelijke overdracht van het kommando aan dBASEIII. Het SCHRM36 wordt getoond en de cursor staat bij de punt, zodat de mogelijkheid bestaat een puntkommando in te voeren. De inhoud van dit kommando hangt van de gebruiker af, maar, als een laatste service, meldt het programma de gebruiker hoe het terug opgeroepen kan worden of hoe men dBASEIII kan verlaten.

PROGRAMMASCHETS.

Dit gedeelte bestaat uit een verzameling lijnschema's. Hierin wordt op een globale wijze aangegeven hoe de verschillende modules aangeroepen worden.

Globaal omdat alle modules worden opgesomd die door een gegeven procedure worden aangeroepen. Dit houdt niet in dat ze ook allen MOETEN opgeroepen worden bij het uitvoeren van het programma. Al naar gelang de voorwaarden

SCHRM36:

Je verliet het bibliotheekprogramma

Je kunt

- of terug naar dit programma
via DO PROG <CR>
- of terug naar dBASE via nnnn
<CR> met nnnn een dBASE
opdracht
- of terug naar DOS via QUIT <CR>

SCHRM36

zal soms de ene, soms de andere module bij de uitvoering betrokken zijn. De opsomming van de variabelen die als parameter worden uitgewisseld is hierin erg belangrijk. Deze vermelding gebeurt op tweeërlei manieren:

1) in de aanroepende procedure worden de mee te geven variabelen opgesomd tussen {}. Binnen de accolades vindt men de opsomming van de variabelen zoals ze genoemd worden in de aanroepende procedure. Deze variabelen kunnen eigenlijk op tweeërlei wijze aan de aangeroepen module overgemaakt worden. Dit kan door refereren gebeuren. In dit geval krijgt de aanroepende procedure een waarde terug van de opgeroepen module. Werd in de module de INHOUD van de variabele gewijzigd, dan komt deze nieuwe waarde in de aanroepende procedure terecht. Indien wijzigingen niet aanvaardbaar zijn, kunnen de variabelen als VALUE parameters overgedragen worden. Het volstaat daartoe bij de oproep de variabele tussen ronde haakjes te plaatsen;

2) in de aangeroepen module worden de variabelen door de PARAMETER opdracht geïnitieerd. In de aangeroepen module zijn ze bekend onder de namen die opgesomd worden tussen [{}]. Ze stemmen uiteraard één voor één met de variabelen uit de vorige paragraaf overeen.

In een dBASEIII+ programma is het niet gebruikelijk om zoveel parameters aan te wenden. Het is evenwel bijzonder gemakkelijk indien vele, kleine modules geschreven worden. Door deze "black-box" benadering is het niet nodig over de verschillende niveau's heen dezelfde variabele rigoureus met dezelfde naam aan te duiden. Ook het doorgronden van het programma en een latere onderhoud worden door deze techniek gemakkelijker gemaakt.

Een laatste voordeel van zo'n programmaschets is de gemakkelijke bepaling van het aantal bestanden dat op een gegeven ogenblik open is. Om onder MS-DOS 3.2 te kunnen draaien kunnen op hetzelfde ogenblik nooit meer dan 20 bestanden open zijn. De modulaire techniek opent nogal wat bestanden zoals het schema BIB900.LYN leert. Bij het uitvoeren van de module SPLIT zijn niet minder dan 9 programmabestanden open! Naar het hoogste niveau toelopen zijn het:

SPLIT
LSSCHRM

IN
MENU06
TANK
MENU07
MENU01
MENU00
PROG

Daarnaast zijn er 5 bestanden in gebruik door MS-DOS. Een controle van de diverse modules leert dat op hetzelfde ogenblik ook nog een gegevensbestand, een hieraan gekoppeld memobestand en vier indexbestanden geopend zijn. Dit brengt het totaal aantal geopende bestanden op 20!

Ieder schema draagt een nummer in de aard van BIB100.LYN. Het eerste cijfer duidt op het niveau. Deze waarde kan gemakkelijk uit het schema worden afgeleid. Het is het aantal vertikale lijnen eventueel vermeerderd met 1 als geen enkele van de aangeroepen modules nog parametervariabelen tussen alleen maar accolades bezit.

Een andere opmerking heeft betrekking op de initialisatie van de "defaultwaarden". Dit wordt behandeld alsof het hier een aangeroepen module betrof waarin de parameteropdracht tot de initialisatie van de tussen [{}] opgesomde variabelen leidt. Dit lijkt een verantwoorde uitbreiding van de gevolgde werkwijze.

PROG

```

— initialiseer defaultwaarden [{KANTLIJN,CONDENS,MERK,NORMAAL,DETPAPON,
    DETPAPUT,PLN,EINDE,VLAG,DELAY_1,SYSDRV,TUS_F,ANDX,DNDX,PNDX,
    SNDX,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,STDF,WRKF,OPBF}]

— indien nodig creëren van file en indexen

— FOUT1 {ERROR()}

— INFO {DELAY_1}

— INIT0 {(STDF),(ANDX),(DNDX),(PNDX),(SNDX),TUS_F,WRKF,OPBF,NFILE,NDX_1,
    NDX_2,NDX_3,EINDE,DELAY_1,}

— MENU00 {TUS_F,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,VLAG,PLN,KANTLIJN,MERK,
    CONDENS,EINDE,WRKF,SYSDRV,DETPAPON,DETPAPUT}

— herstel opstartwaarden

```

SLUIT

```

—
—
— FOUT1 [{FOUT}]
— INFO [{DELAY_1}]
— INIT0 [{STDF,ANDX,DNDX,PNDX,SNDX,TUS_F,WRKF,OPBF,NFILE,NDX_1,NDX_2,
    NDX_3,NDX_4,EINDE,DELAY_1}]
    — EENANT {NFILE,DUMMY,KOM2,DUMMY,LENGTE}
    — NDXEN {NFILE,NDX_1,NDX_2,NDX_3,NDX_4}
— MENU00 [{TUS_F,NFILE,NDX_1,NDX_2,NDX_4,VLAG,PLN,KANTLIJN,MERK,CONDENS,
    EINDE,WRKF,SYSDRV,PAPON,PAPUT}]
    — EXIT
    — MENU01 {TUS_F,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,VLAG,PLN,KANTLIJN,
        MERK,CONDENS}
    — MENU02 {TIT,K1,K2,K3,K4,K5,K6,K7,K8,FLOPPY,WRKF,NFILE,NDX_1,
        NDX_2,NDX_3,NDX_4,PLN,PAPON,PAPUT,VLAG,KANTLIJN,MERK,
        CONDENS}
    — MENU03 {NFILE,NDX_1,NDX_2,NDX_3,NDX_4,PLN,PAPON,PAPUT,VLAG,
        KANTLIJN,MERK,CONDENS,WRKF}
    — MENU04 {NFILE,NDX_1,NDX_2,NDX_3,NDX_4,PLN,PAPON,PAPUT,VLAG,
        KANTLIJN,MERK,CONDENS,WRKF}
    — MENU05 {NFILE,NDX_1,NDX_2,NDX_3,NDX_4,SYSDRV,PLN,PAPON,PAPUT,
        MERK,KANTLIJN,CONDENS,VLAG}

```

INITO

```

  EENANT [{V1,V2,C1,C2,N}]
  NDXEN [{AFILE,NDX_1,NDX_2,NDX_3,NDX_4}]

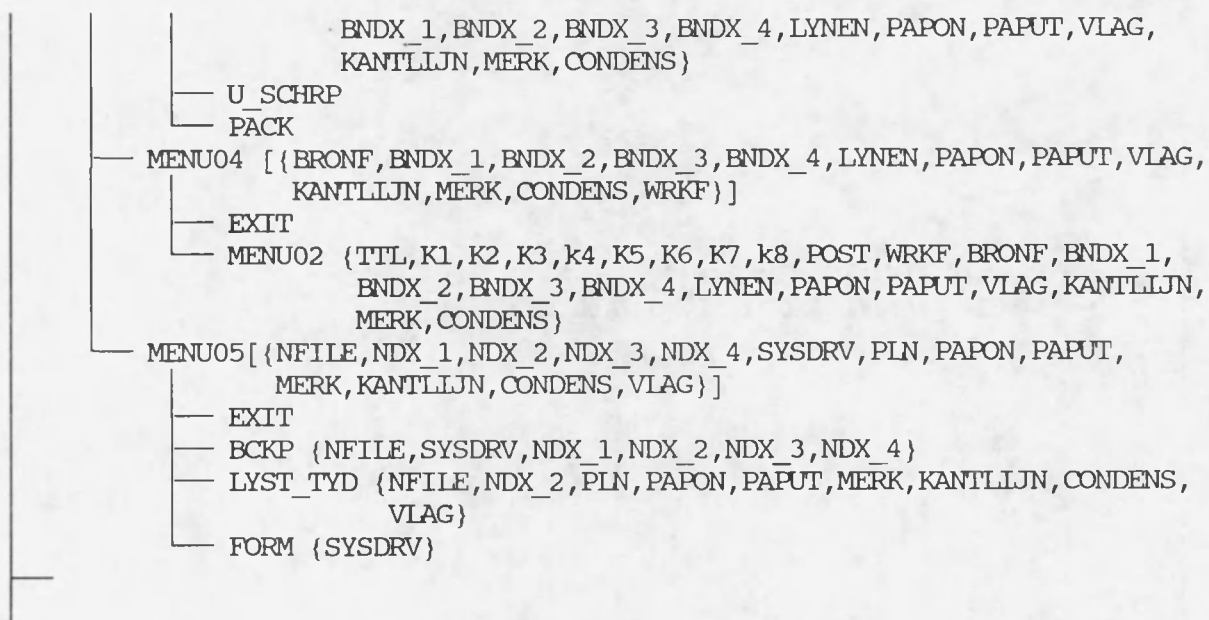
```

MENU00

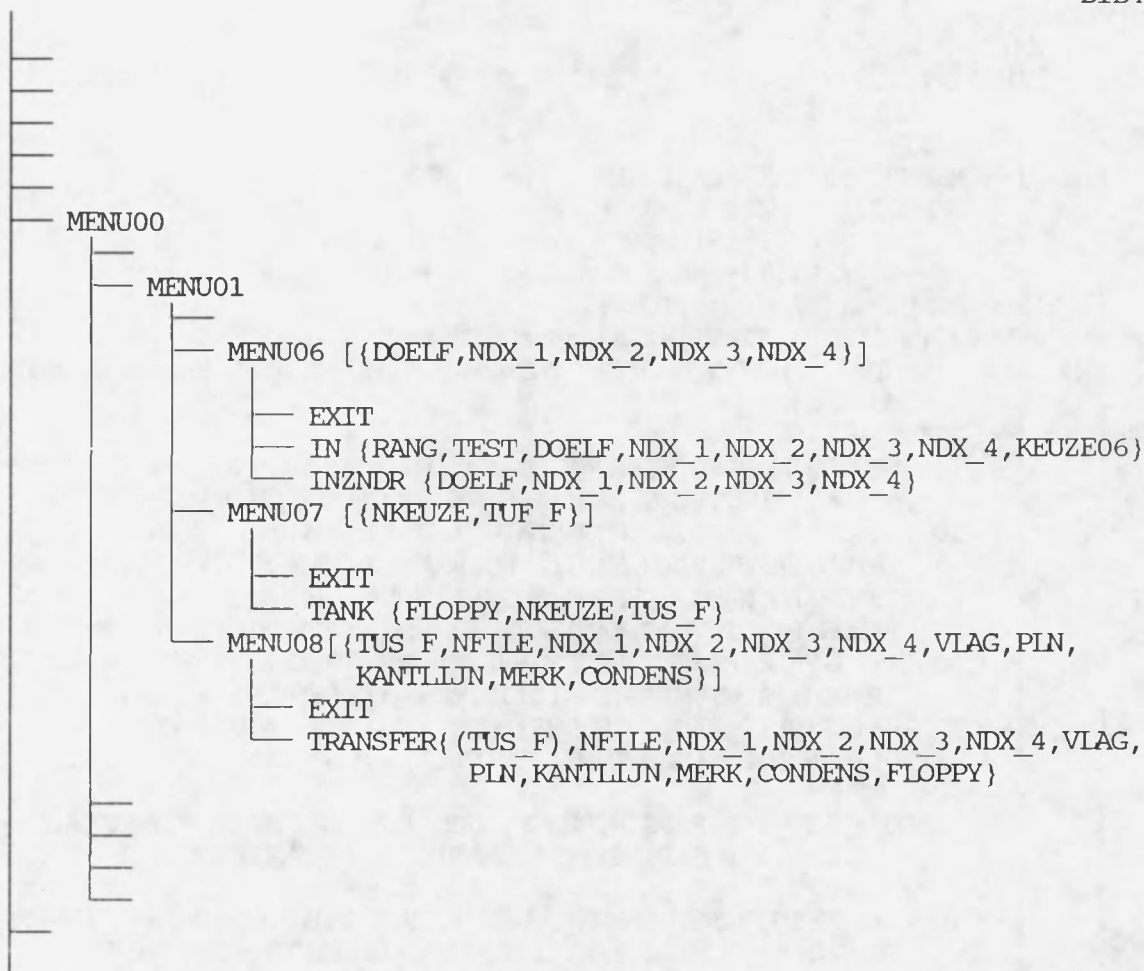
```

  EXIT
  MENU01 [{TUS_F,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,VLAG,PLN,KANTLLJN,
    MERK,CONDENS}]
    EXIT
    MENU06 {NFILE,NDX_1,NDX_2,NDX_3,NDX_4}
    MENU07 {(KEUZE01),TUS_F}
    MENU08 {TUS_F,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,VLAG,PLN,
      KANTLLJN,MERK,CONDENS}
    MENU07 {(KEUZE01),TUS_F}
  MENU02 [{TTITEL,K1,K2,K3,K4,K5,K6,K7,K8,FLOPPY,WRKF,BRONF,BNDX_1,
    BNDX_2,BNDX_3,BNDX_4,LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,
    CONDENS}]
    EXIT
    IO {TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,LYNEN,
      PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,WRKF}
    IO {TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,LYNEN,
      PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,WRKF}
    IO {TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,LYNEN,
      PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,WRKF}
    IOT {TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,LYNEN,
      PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,WRKF}
    IONR {TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,
      LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,
      WRKF}
    IONRS {TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,
      LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,
      WRKF}
    IO {TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,LYNEN,
      PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,WRKF}
    IOAND{TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,
      LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,
      WRKF}
  MENU03 [{BRONF,BNDX_1,BNDX_3,BNDX_4,BNDX_4,LYNEN,PAPON,PAPUT,VLAG,
    KANTLLJN,MERK,CONDENS,WRKF}]
    EXIT
    MENU02 {TTITEL,K1,K2,K3,K4,K5,K6,K7,K8,FLOPPY,WRKF,BRONF,

```



BIB400.LYN



MENU00

MENU02

IO [{KEUZE,RANG,N,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,LYNEN,PAPON,
 PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,WRKF}]
 — EENANT {KRIT1,DUMMY,KOM3,KOM2,N}
 — KOMTR_1 {KOM1,KOL1,LEERSTE,LLAATSTE}
 — LSSCHRM
 — WISSEL_1 {ANT,GEW,REC,STOPPEN,MAUTEUR,MITTEL,MDRAGER,
 MPLAATS,MAARD,MTREFWRD,RANG}
 — LGSCHRM
 — OPDR601 {KRIT1}

IO

IO

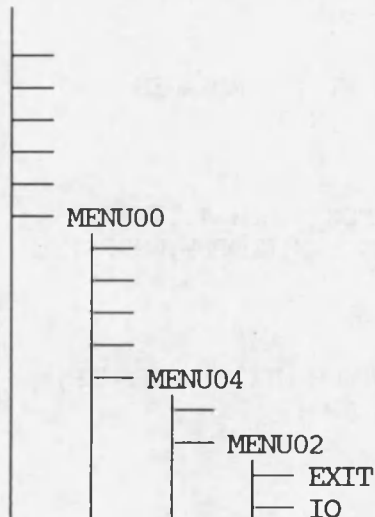
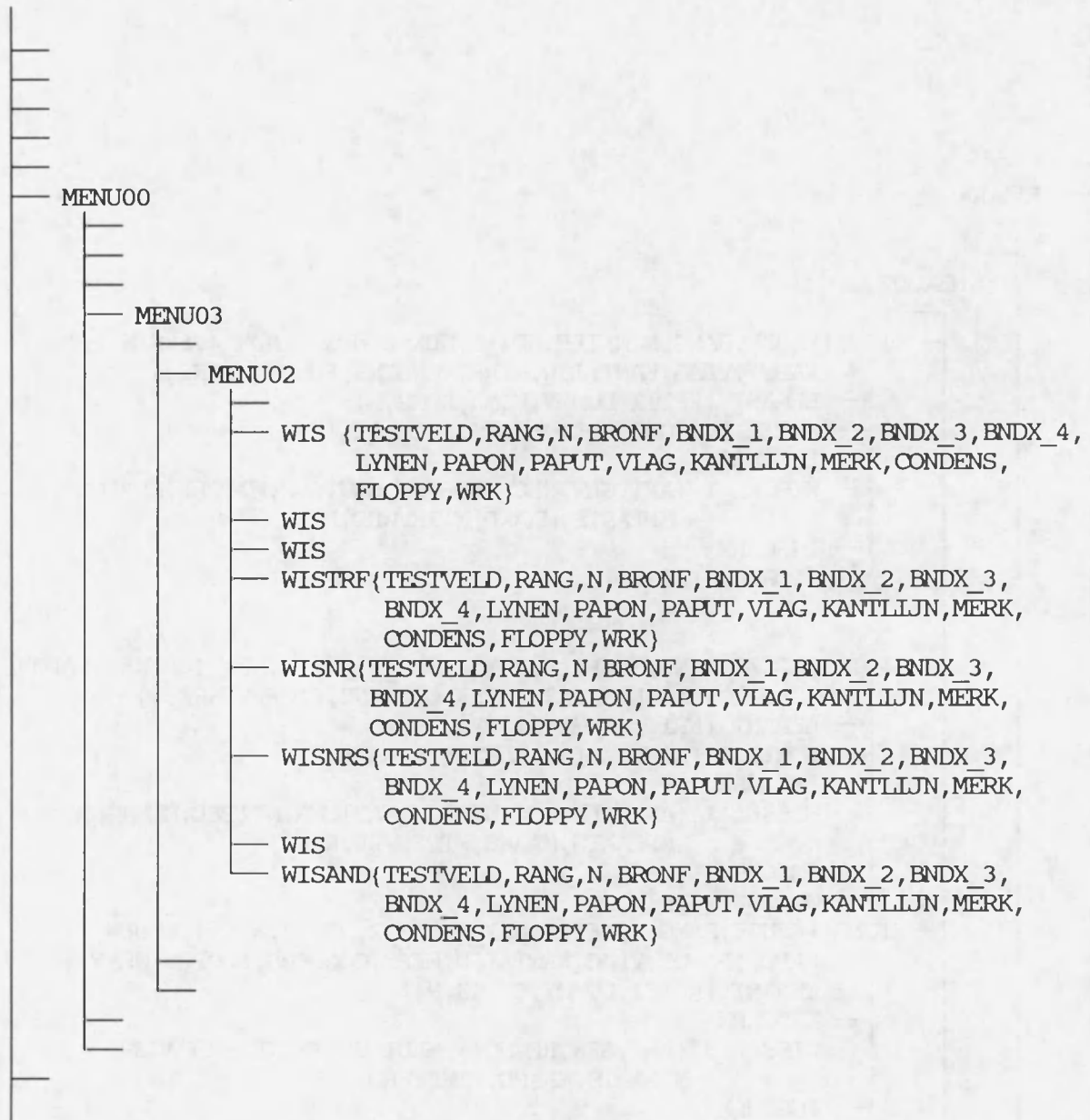
IOT [{KEUZE,RANG,N,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,LYNEN,PAPON,
 PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,WRKF}]
 — EENANT {KRIT,DUMMY,C1,C2,N}
 — KOMTR_1 {KOM1,KOL1,LEERSTE,LLAATSTE}
 — LSSCHRM
 — WISSEL_1 {ANT,GEW,REC,NIDRGN,MAUTEUR,MITTEL,MDRAGER,
 MPLAATS,MAARD,MTREFWRD,RANG}
 — LGSCHRM
 — OPDR601 {KRIT}

IONR [{KEUZE,RANG,N,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,LYNEN,
 PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,WRKF}]
 — EENANT {KRIT1,DUMMY,C1,C3,N}
 — LSSCHRM
 — WISSEL_3 {ANT,GEW,RECNO(),MAUTEUR,MITTEL,MDRAGER,
 MPLAATS,MAARD,MTREFWRD,RANG}
 — LGSCHRM

IONRS [{KEUZE,RANG,N,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,LYNEN,
 PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,WRKF}]
 — TWEEANT {NR1,NR2,C1,C2,C3}
 — KOMTR_1 {KOM2,KOL2,LEERSTE,LLAATSTE}
 — LSSCHRM
 — WISSEL_2 {ANT,GEW,REC,RANG,MAUTEUR,MITTEL,MDRAGER,
 MPLAATS,MAARD,MTREFWRD,NR1,NR2}
 — LGSCHRM

IO

IOAND [{KEUZE,RANG,N,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,LYNEN,
 PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,WRKF}]
 — OPDR604
 — KOMTR_1 {KOM1,KOL,LEERSTE,LLAATSTE}
 — LSSCHRM
 — WISSEL_1 {ANT,GEW,REC,STOPPEN,MAUTEUR,MITTEL,MDRAGER,
 MPLAATS,MAARD,MTREFWRD,RANG}
 — LGSCHRM
 — OPDR601 {KRIT1}



```

— IO
— IO
— IOT
— IONR
— IONRS
— IO
— IOAND
— MENU02
— EXIT
— PRT {TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,
      LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,
      WRKF}
— PRT
— PRT
— PTR {TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,
      LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,
      WRKF}
— PNR {TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,
      LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,
      WRKF}
— PNRS {TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,
      LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,
      FLOPPY,WRKF}
— PRT
— PAND {TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,
      LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,
      FLOPPY,WRKF}
— MENU02
— EXIT
— COPIE {TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,
      BNDX_4,LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,
      CONDENS,FLOPPY,WRKF}
— COPIE
— COPIE
— CTRF {TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,
      LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,
      FLOPPY,WRKF}
— CNR {TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,
      LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,
      WRKF}
— CNRS {TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,
      LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,
      FLOPPY,WRKF}
— COPIE
— CAND {TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,
      LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,
      FLOPPY,WRKF}

```

MENU00

MENU05

```

BCKP [ { BRONF, SYSDRV, NDX_1, NDX_2, NDX_3, NDX_4 } ]
  |
  | EENANT { POST, DUMMY, KOM2, KOM3, LENGTE }
  | EENANT { POST, DUMMY, KOM1, DUMMY, LENGTE }
  |
  | LYST_TYD [ { BRONF, BNDX, LYNEN, PAPON, PAPUT, MERK, KANTLLJN,
  |             CONDENS, VLAG } ]
  | INSTR { VLAG, LYNEN, KANTLLJN, MERK, CONDENS }
  |
  | FORM [ { SYSDRV } ]

```

MENU00

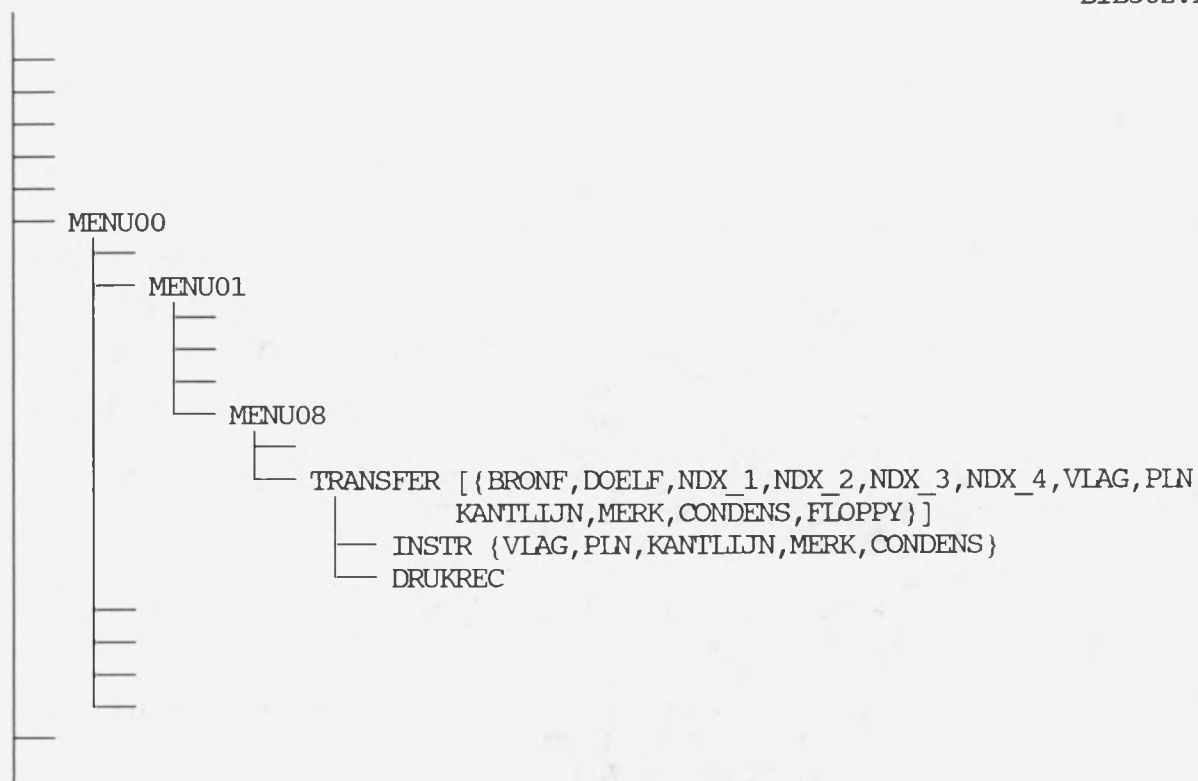
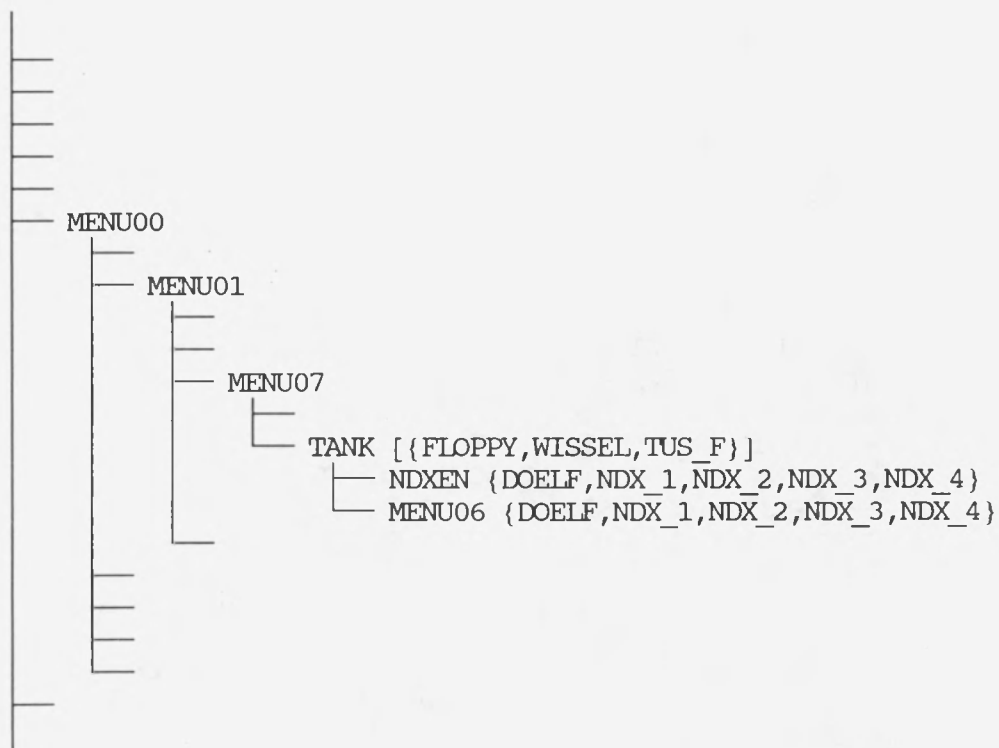
MENU01

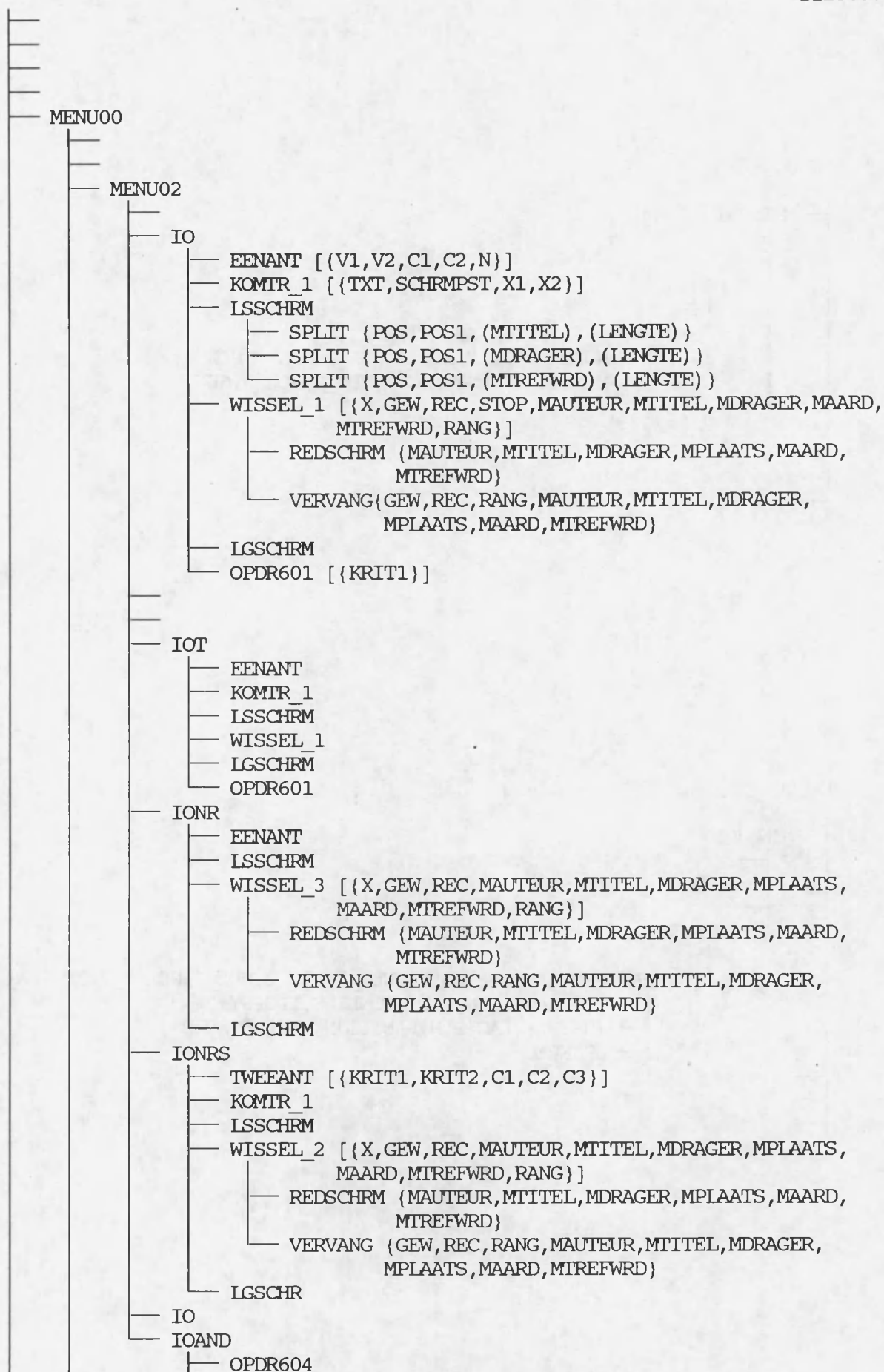
MENU06

```

IN [ { RANG, TEST, DOELF, NDX_1, NDX_2, NDX_3, NDX_4, KEUZE } ]
  |
  | EENANT { KRIT1, KRIT0, KOM3, KOM, N }
  | LSSCHRM
  | INSCHRM { KEUZE, MAUTEUR, MTITEL, MDRAGER, MPLAATS,
  |           MIREFWRD, KRIT1 }
  |
  | INZNDR [ { DOELF, NDX_1, NDX_2, NDX_3, NDX_4, KEUZE } ]
  | INSCHRM { KEUZE, MAUTEUR, MTITEL, MDRAGER, MPLAATS,
  |           MIREFWRD, KRIT1 }

```





- KOMIR_1
- LSSCHRM
- WISSEL_1
- LGSCHRM
- OPDR601

BIB504.LYN

MENU00

MENU03

MENU02

- WIS [{ KEUZE, RANG, N, BRONF, BNDX_1, BNDX_2, BNDX_3, BNDX_4, LYNEN, PAPON, PAPUT, VLAG, KANTILLJN, MERK, CONDENS, WERKF }]
 - EENANT { KRIT, DUMMY, C1, C2, N }
 - KOMIR_1 { KOM1, KOL1, LEERSTE, LLAATSTE }
 - LSSCHRM
 - WISSEL_4 { ANT, REC, STOPPEN, LEERSTE, LLAATSTE }
 - OPDR601 { KRIT }
- WIS
- WIS
- WISTRF [{ KEUZE, RANG, N, BRONF, BNDX_1, BNDX_2, BNDX_3, BNDX_4, LYNEN, PAPON, PAPUT, VLAG, KANTILLJN, MERK, CONDENS, FLOPPY, WRKF }]
 - EENANT { KRIT, DUMMY, C1, C2, N }
 - KOMIR_1 { KOM1, KOL1, LEERSTE, LLAATSTE }
 - LSSCHRM
 - WISSEL_4 { ANT, REC, STOPPEN, LEERSTE, LLAATSTE }
 - OPDR601 { KRIT }
- WISNR [{ KEUZE, RANG, N, BRONF, BNDX_1, BNDX_2, BNDX_3, BNDX_4, LYNEN, PAPUT, PAPON, VLAG, KANTILLJN, MERK, CONDENS, FLOPPY, WRKF }]
 - EENANT { KRIT, DUMMY, C1, C2, N }
 - LSSCHRM
- WISNRS [{ KEUZE, RANG, N, BRONF, BNDX_1, BNDX_2, BNDX_3, BNDX_4, LYNEN, PAPUT, PAPON, VLAG, KANTILLJN, MERK, CONDENS, FLOPPY, WRKF }]
 - TWEEANT { NR1, NR2, C1, C2, C3 }
 - KOMIR_1 { KOM2, KOL2, LEERSTE, LLAATSTE }
 - LSSCHRM
- WIS
- WISAND [{ KEUZE, RANG, N, BRONF, BNDX_1, BNDX_2, BNDX_3, BNDX_4, LSSCHRM }

— WISSEL 4 {ANT,REC,STOPPEN,LEERSTE,LLAATSTE}
 — OPDR601 {KRIT}

BIB505.LYN

MENU00

MENU04

MENU02

— PRT [{KEUZE,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,
 LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,
 WRKF}]
 — INSTR {VLAG,LYNEN,KANTLLJN,MERK,CONDENS}
 — EENANT {KRIT,DUMMY,C1,C2,N}
 — BLAD {BLZ,KEUZE,KRIT,LYNEN,PAPON,PAPUT}
 — OPDR601 {KRIT}

— PRT
 — PRT
 — PTR [{KEUZE,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,
 LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,
 WRKF}]
 — INSTR {VLAG,LYNEN,KANTLLJN,MERK,CONDENS}
 — EENANT {KRIT,DUMMY,C1,C2,N}
 — BLAD5 {BLZ,KEUZE,KRIT,LYNEN,PAPON,PAPUT}
 — OPDR601 {KRIT}

— PNR [{KEUZE,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,
 LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,
 WRKF}]
 — INSTR {VLAG,LYNEN,KANTLLJN,MERK,CONDENS}
 — EENANT {KRIT,DUMMY,C1,C2,N}
 — BDSCHP2

— PNRS [{KEUZE,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,
 LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,
 FLOPPY,WRKF}]
 — INSTR {VLAG,LYNEN,KANTLLJN,MERK,CONDENS}
 — TWEEANT {NR1,NR2,C1,C2,C3}
 — BLAD4 {BLZ,LYNEN,NR1,NR2,PAPON,PAPUT,INR1,INR2}

— PRT
 — PAND [{KEUZE,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,
 LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,
 FLOPPY,WRKF}]

- INSTR {VLAG,LYNEN,KANTLIJN,MERK,CONDENS}
- OPDR604
- BLAD2 {BLZ,KOM3,LYNEN,PAPON,PAPUT}
- OPDR601 {KRIT}

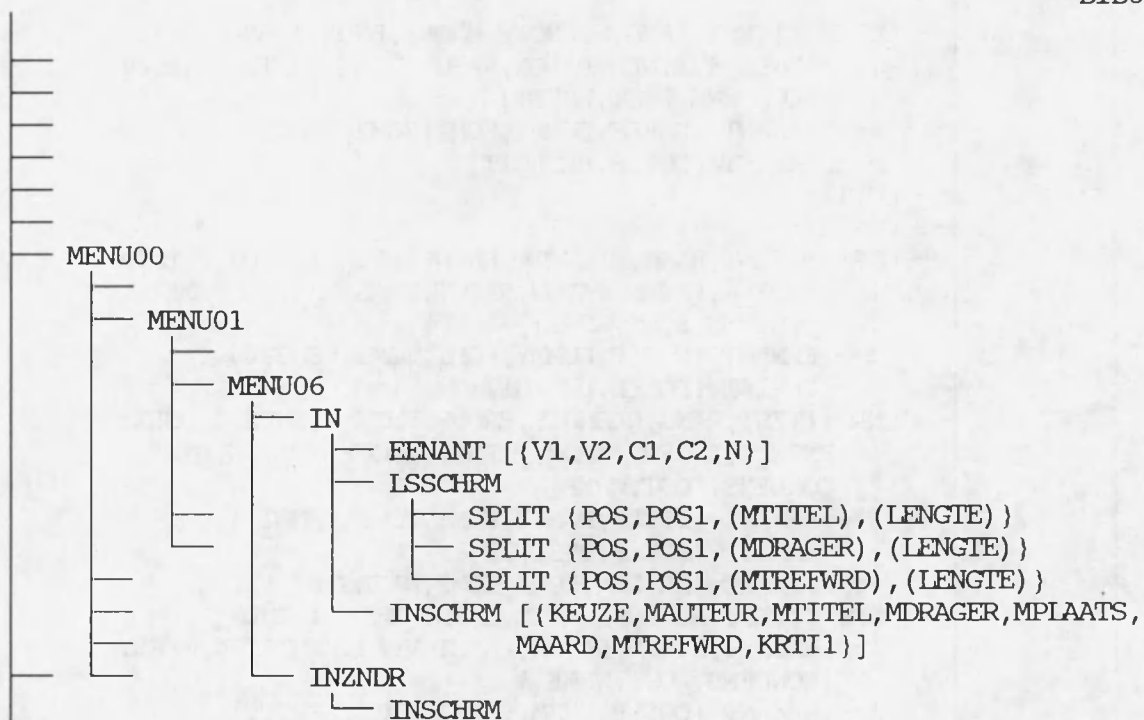
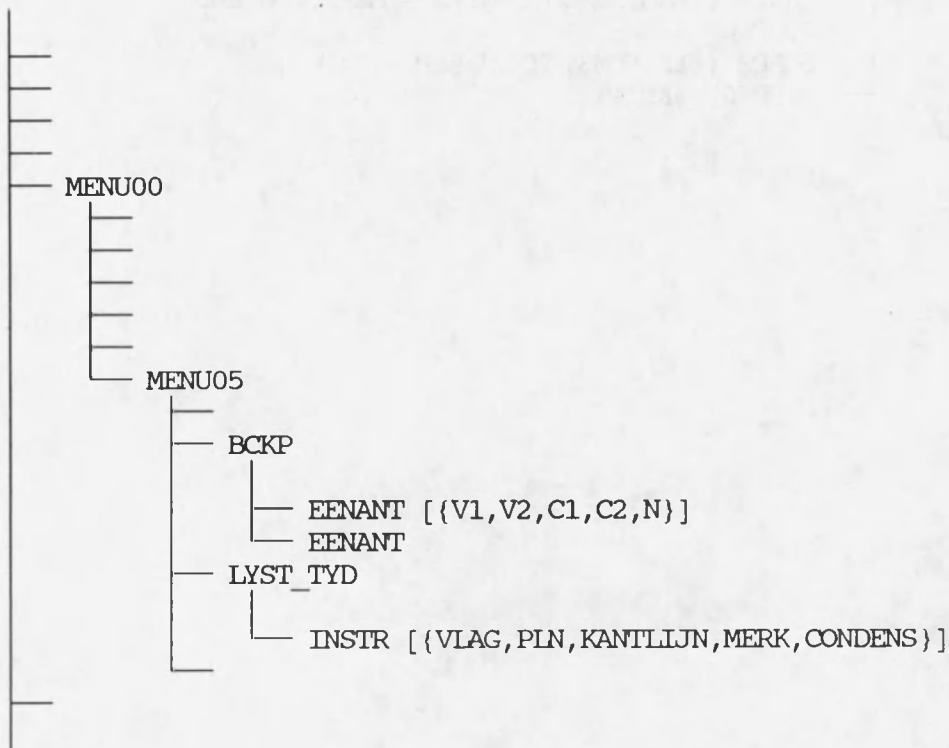
BIB506.LYN

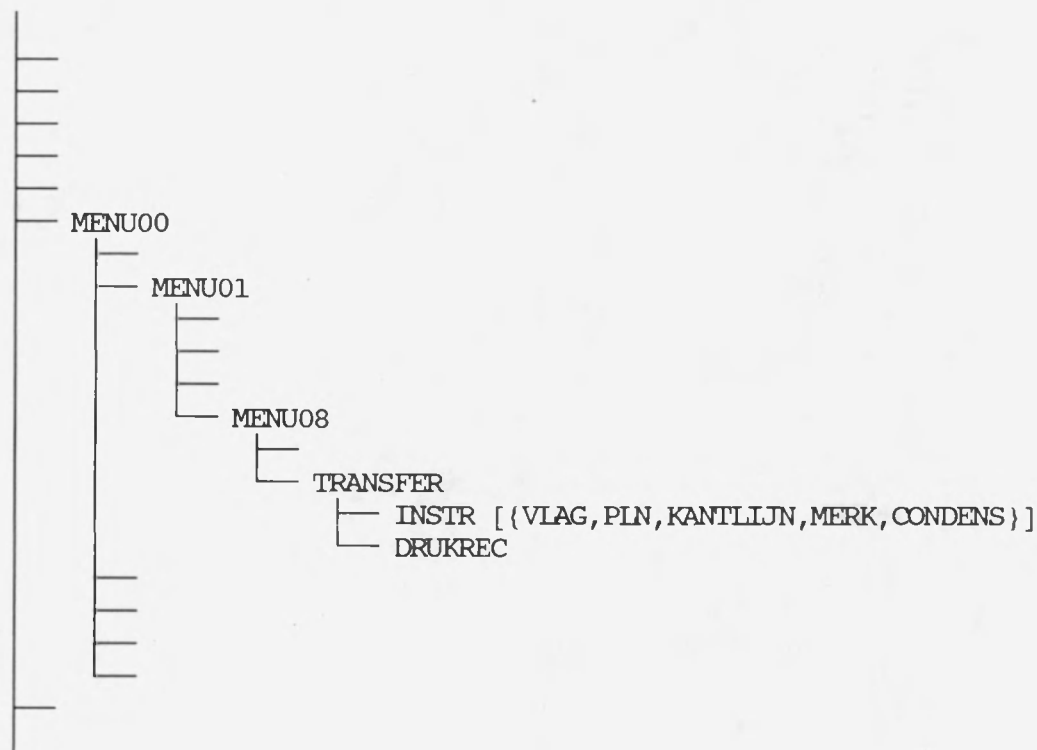
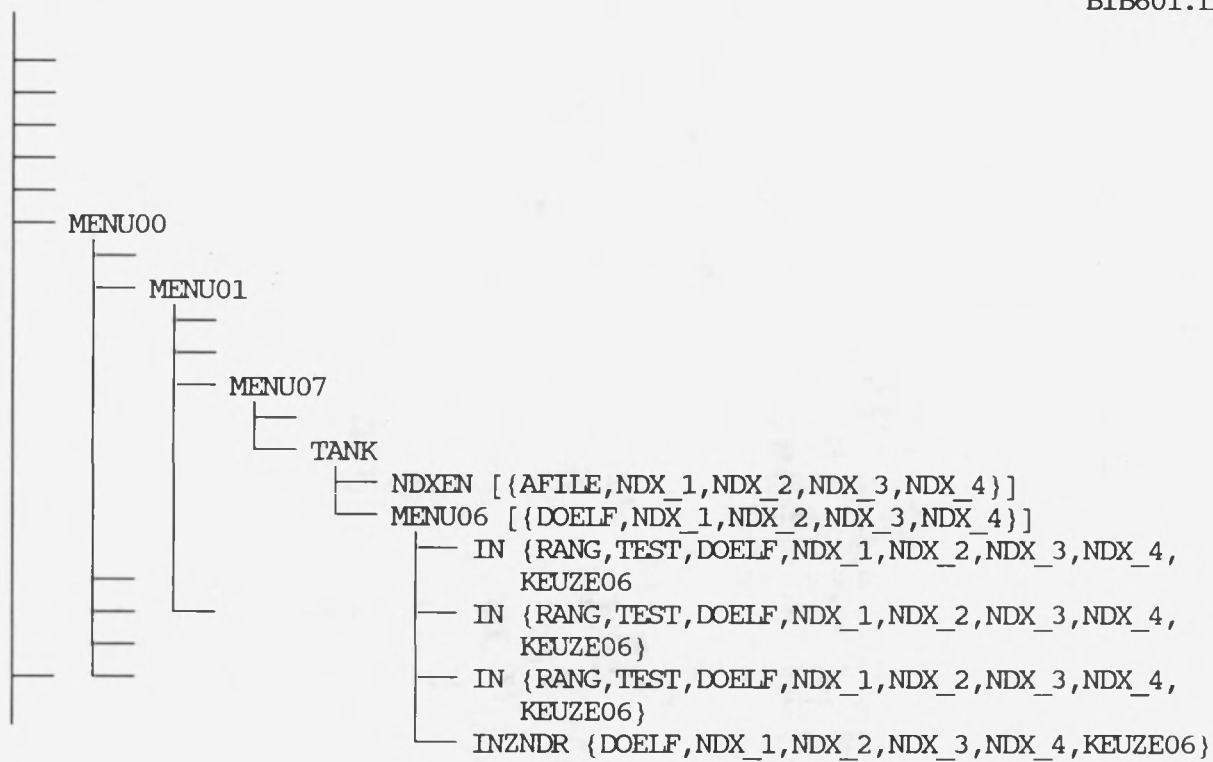
MENU00

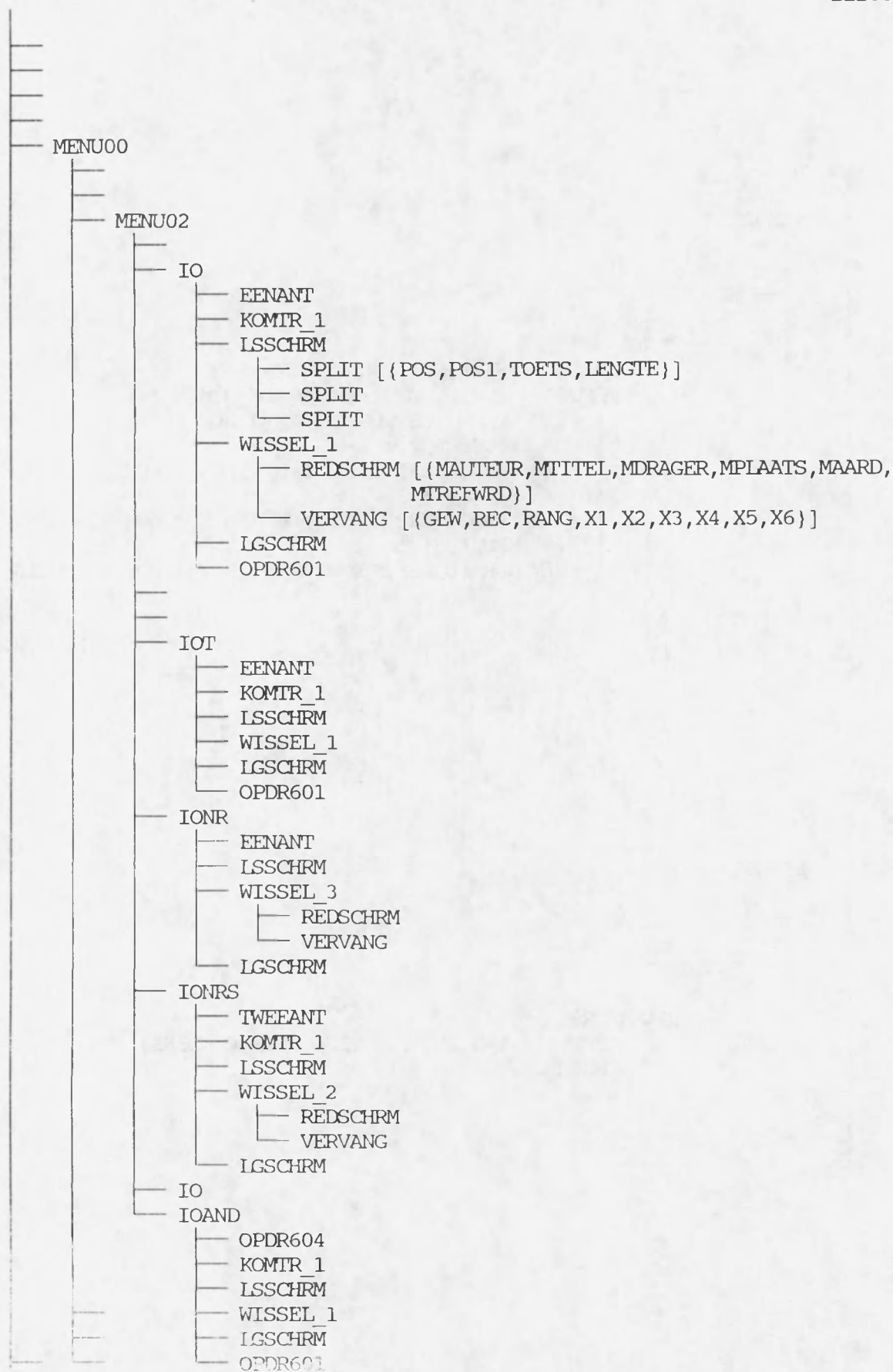
MENU04

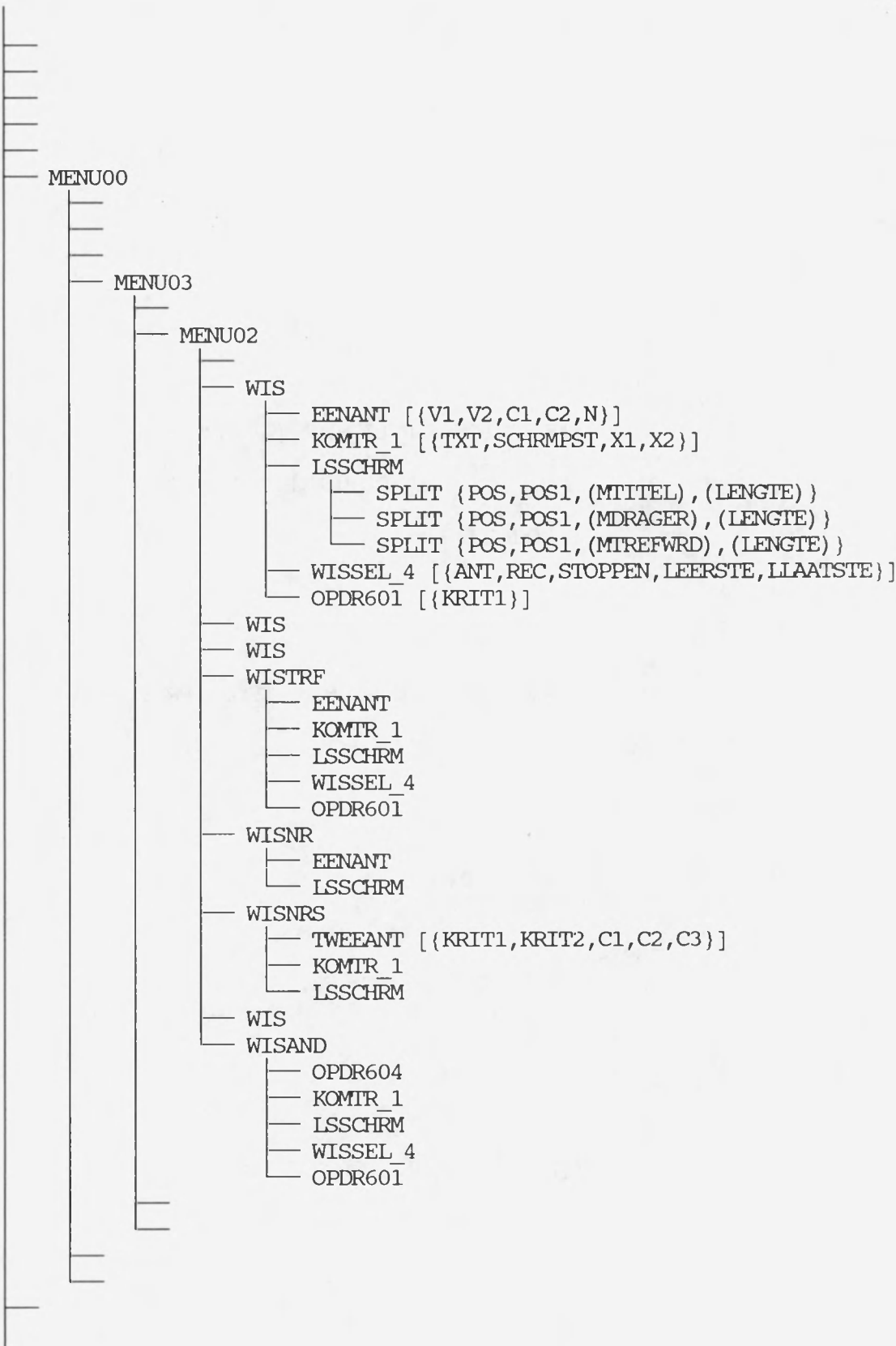
MENU02

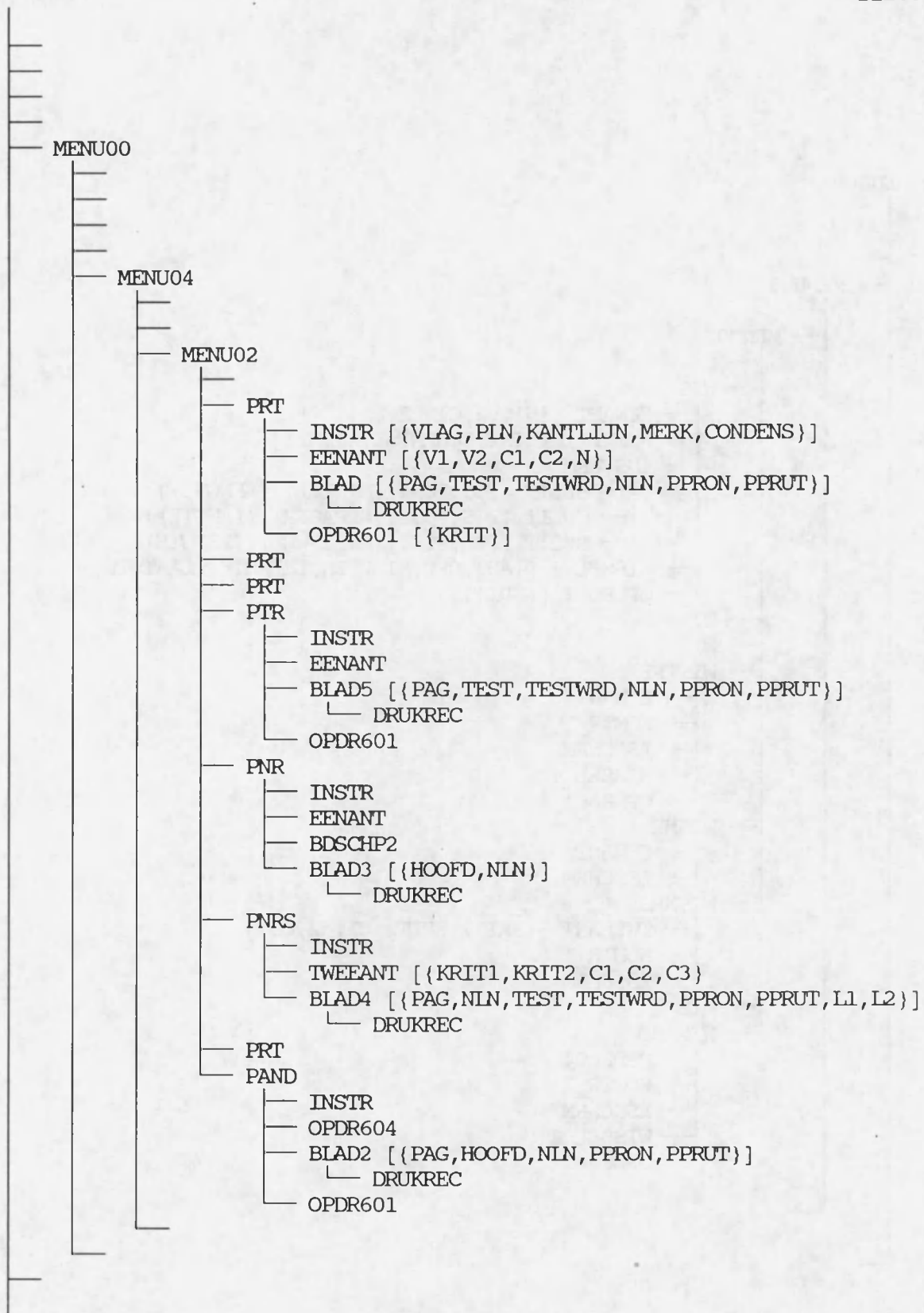
- COPIE [{TEST,RANG,NLENGTE,BRONF,BNDX_1,BNDX_2,BNDX_3,
BNDX_4,LYNEN,PAPON,PAPUT,VLAG,KANTLIJN,MERK,
CONDENS,POST,WRKF}]
 - EENANT {DOELF,DUMMY,KOM1,KOM2,LENGTE}
 - FEXIST {DOELF,BESLUIT}
- COPIE
- COPIE
- CTRF [{TEST,RANG,NLENGTE,BRONF,BNDX_1,BNDX_2,BNDX_3,
BNDX_4,LYNEN,PAPON,PAPUT,VLAG,KANTLIJN,MERK,
CONDENS,POST,WRKF}]
 - EENANT {DOELF,DUMMY,KOM1,KOM2,LENGTE}
 - FEXIST {DOELF,BESLUIT}
- CNR [{TEST,RANG,NLENGTE,BRONF,BNDX_1,BNDX_2,BNDX_3,
BNDX_4,LYNEN,PAPON,PAPUT,VLAG,KANTLIJN,MERK,
CONDENS,POST,WRKF}]
 - EENANT {DOELF,DUMMY,KOM1,KOM2,LENGTE}
 - FEXIST {DOELF,BESLUIT}
 - EENANT {NR,DUM,KOM4,KOM2,NLENGTE}
- CNRS [{TEST,RANG,NLENGTE,BRONF,BNDX_1,BNDX_2,BNDX_3,
BNDX_4,LYNEN,PAPON,PAPUT,VLAG,KANTLIJN,MERK,
CONDENS,POST,WRKF}]
 - EENANT {DOELF,DUMMY,KOM1,KOM2,LENGTE}
 - FEXIST {DOELF,BESLUIT}
 - TWEEANT {NR1,NR2,KOM2,KOM3,KOM4}
- CAND [{TEST,RANG,NLENGTE,BRONF,BNDX_1,BNDX_2,BNDX_3,
BNDX_4,LYNEN,PAPON,PAPUT,VLAG,KANTLIJN,MERK,
CONDENS,POST,WRKF}]
 - EENANT {DOELF,DUMMY,KOM1,KOM2,LENGTE}
 - FEXIST {DOELF,BESLUIT}
 - OPDR604
 - OPDR601 {KRIT1}

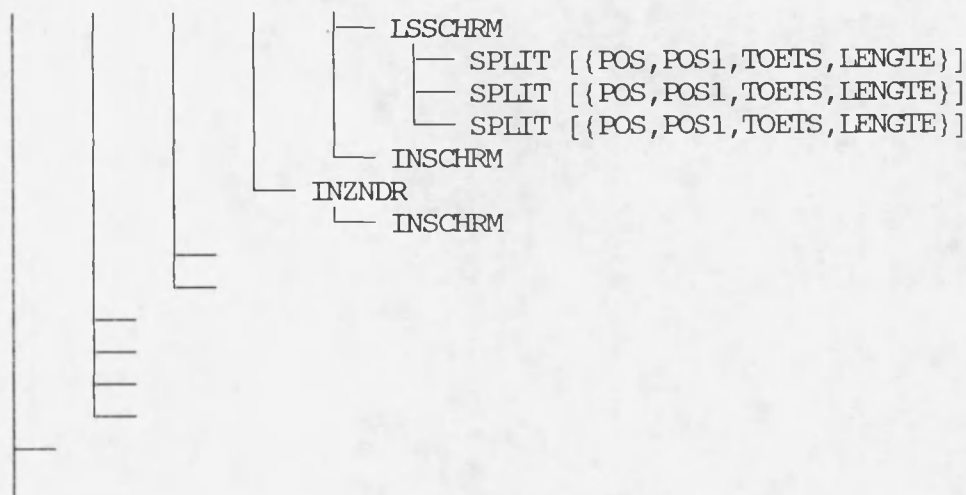




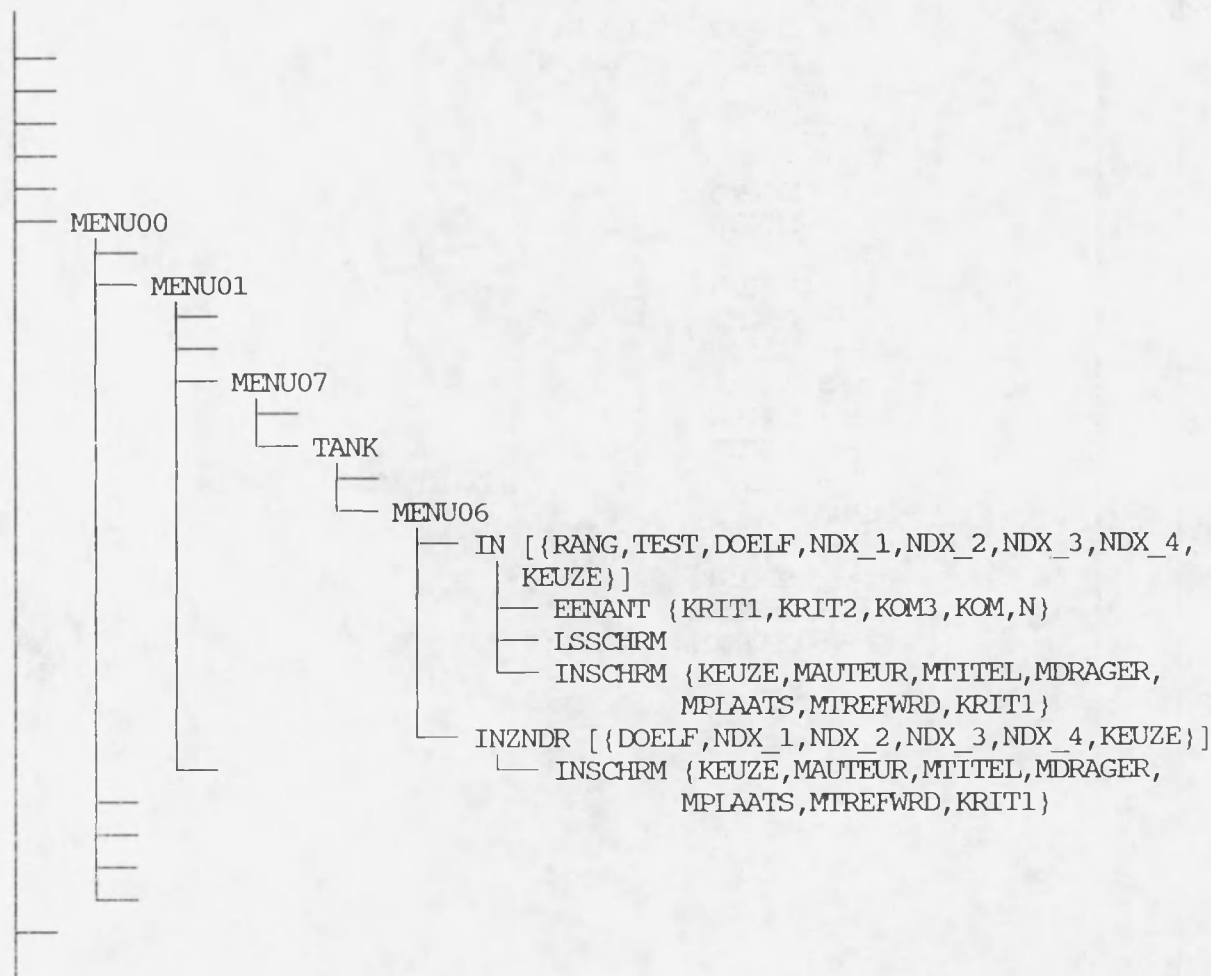


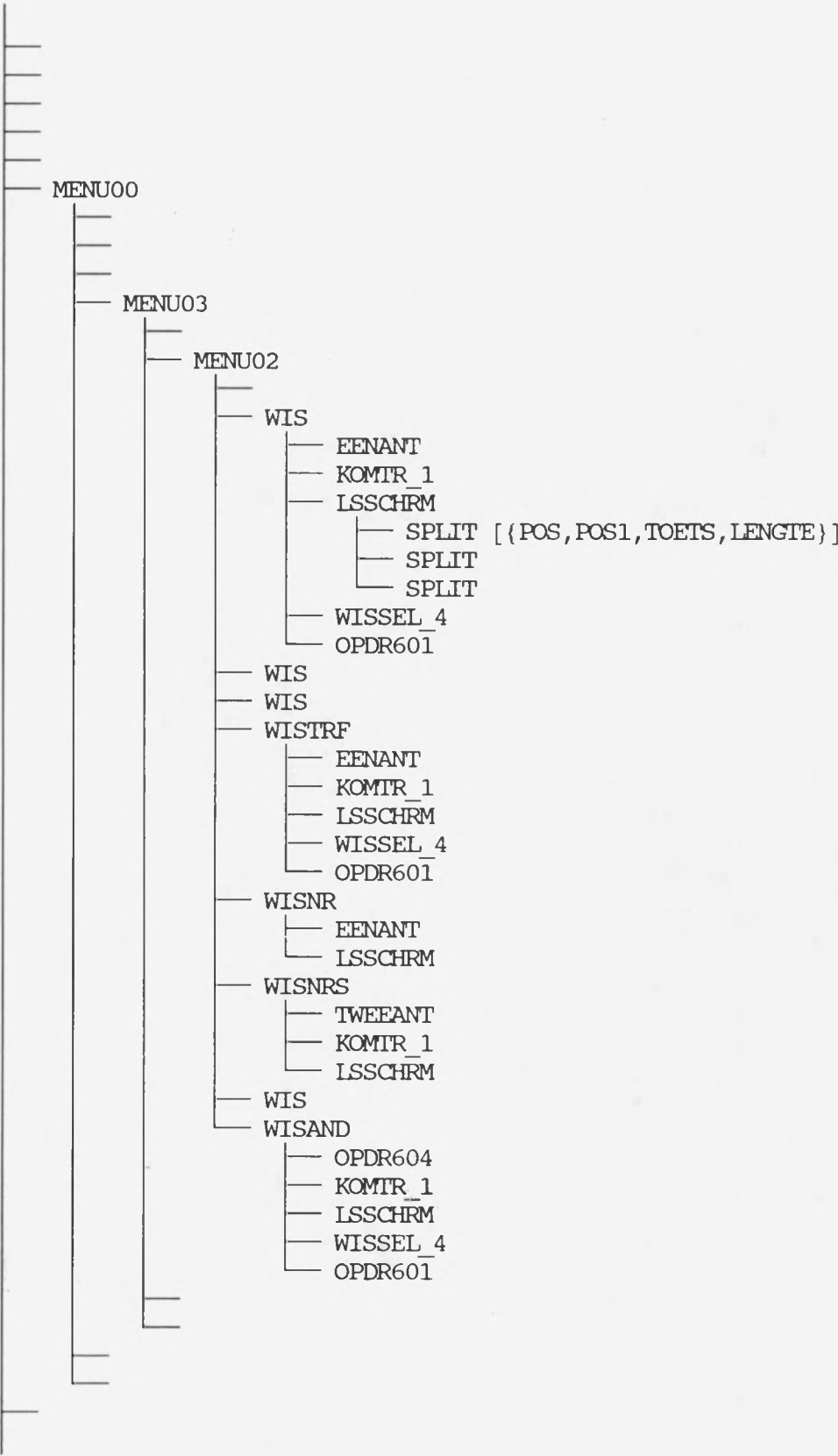


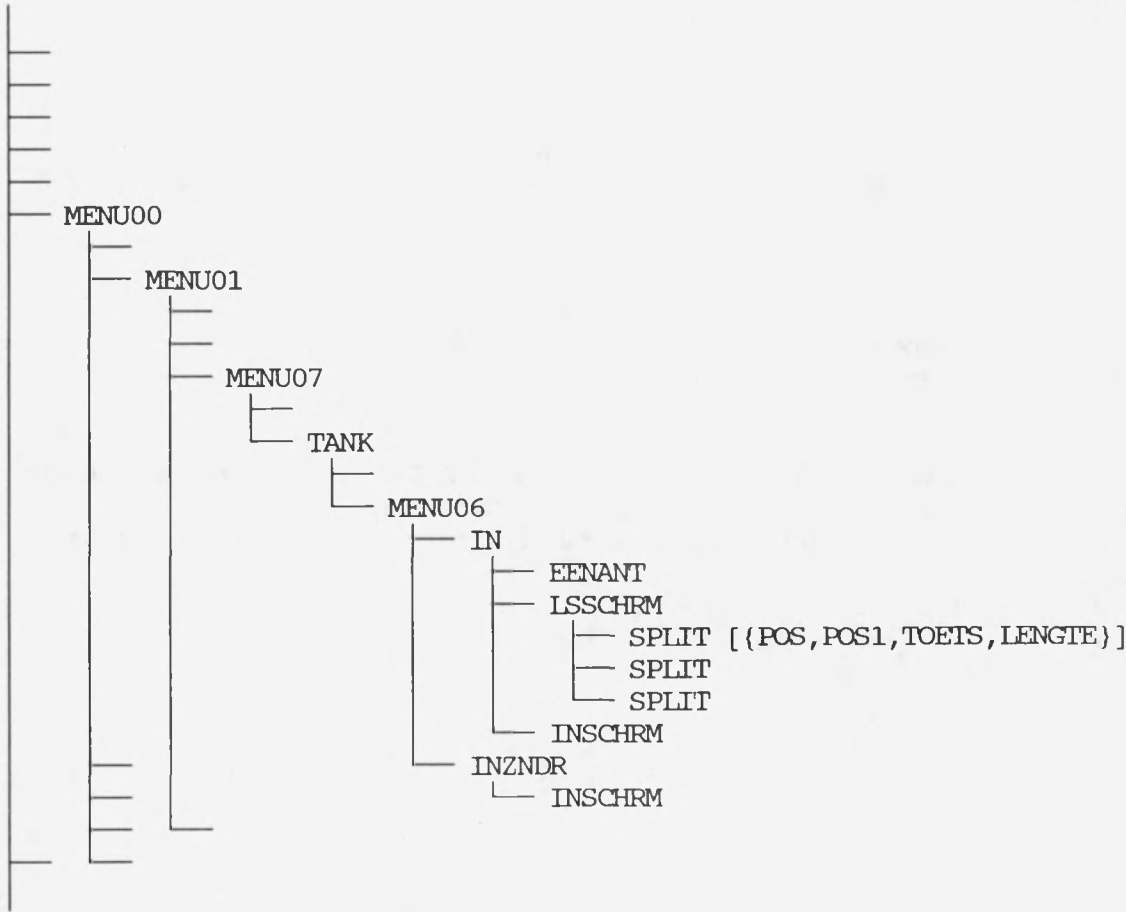
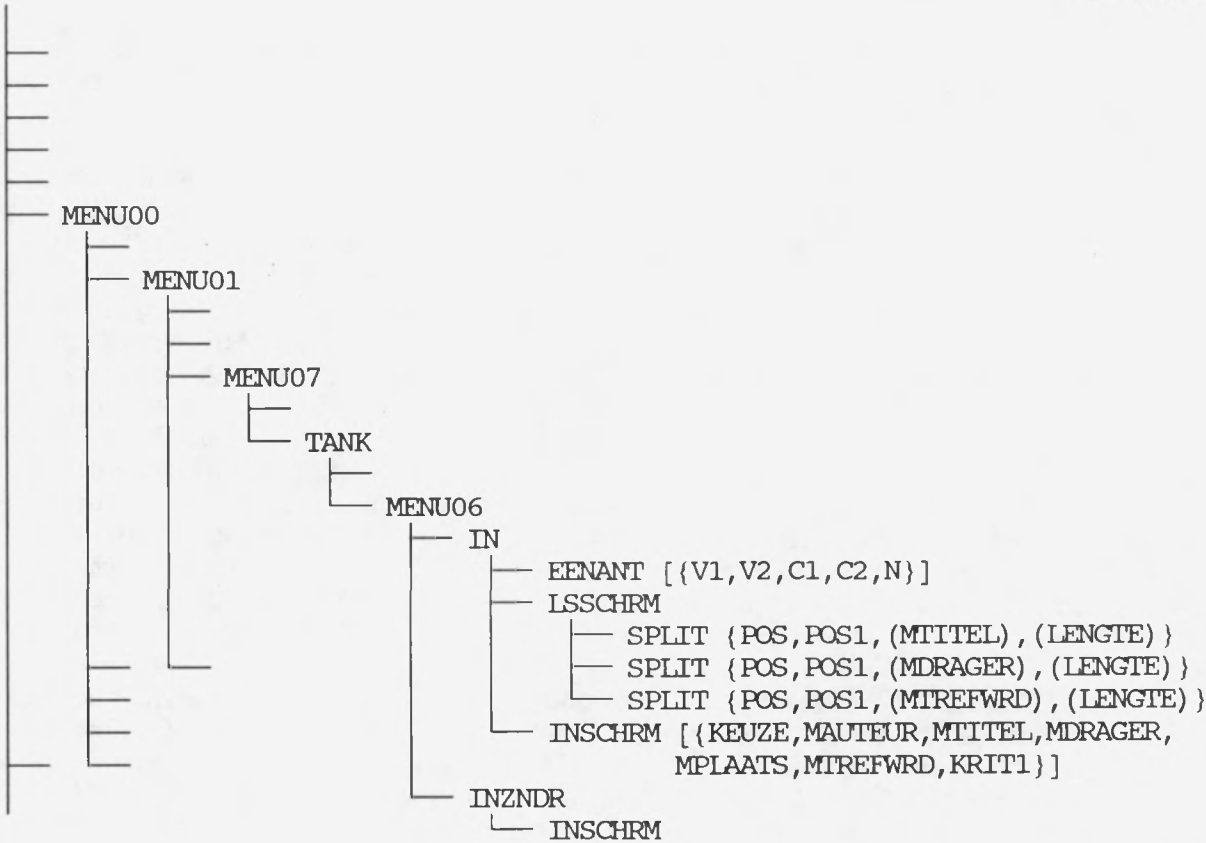




BIB701.LYN







DE PROGRAMMAMODULES.

De modules vormen het programma. Ze worden alfabetisch volgens hun roepnaam weergegeven. Er is een korte verklaring in verband met de rol van de module.

Bij het installeren van het programma is het niet voldoende de verschillende modules foutloos in te typen. Het programma moet nog beschikken over een WAARDEN.mem, een OPBOUW.dbf en een WRKF_1.dbf bestand.

Het aanmaken van een WAARDEN.mem bestand onder dBASEIII+ is geen kunst. Nadat het openingsscherm van dBASEIII+ vervangen werd door het ASSIST menu wordt de ESC toets ingedrukt. Het ASSIST menu verdwijnt en dBASEIII+ bevindt zich in de "puntkommando" toestand. De namen van de variabelen en hun inhoud worden ingetypt en nadat het laatste gegeven ingevoerd werd, wordt het kommando SAVE TO WAARDEN.mem ingetoetst. Dient een bestaand WAARDEN.mem bestand te worden aangepast, dan kan na het verdwijnen van het ASSIST menu het kommando RESTORE FROM WAARDEN.mem ingevoerd worden. Het indrukken van de F6 toets brengt de inhoud van het bestand op het scherm. Na het aanpassen van de inhoud van bepaalde variabelen kan het kommando SAVE TO WAARDEN.mem worden uitgevoerd.

Het OPBOUW.dbf bestand wordt gemaakt met behulp van het ASSIST menu. Hierin wordt eerst een databasebestand aangemaakt met een willekeurige naam. De veldnamen en de veldlengtes dienen uiteraard konform te zijn aan de gegevens die op pagina 2 werden vermeld. Nadat het laatste veld is gedefinieerd (het memoveld) wordt het ontworpen databasebestand verlaten, het ASSIST menu wordt verlaten en via het "puntkommando"

COPY STRUCTURE EXTENDED TO OPBOUW.DBF

wordt het bestand aangemaakt.

Op analoge wijze wordt het bestand WRKF_1.dbf aangemaakt. Hiervoor wordt slechts één veld gedefinieerd waarvoor de volgende gegevens werden gebruikt:

veldnaam	RECNR
type	numeriek
lengte	5
aantal decimalen	0

Via het kommando QUIT wordt vervolgens dBASEIII+ verlaten en met een geschikte methode worden zonodig het WAARDEN.mem, het OPBOUW.dbf en het WRKF_1dbf bestand gekopieerd naar de subdirectory waarin de programmamodules opgeslagen zijn.

```

***** BCKP.PRG
*
PARAMETERS BRONF,SYSDRV,NDX_1,NDX_2,NDX_3,NDX_4
CLEAR
DOELF = BRONF
STORE ' ' TO POST,DUMMY
KOM1 = 'brondrive...      (STOPPEN: ESC) '
KOM2 = 'doeldrive...      (STOPPEN: ESC) '
KOM3 = ''
KOM4 = 'Backup-Menu'
KOM5 = 'plaats MS-DOS diskette in A:'
KOM6 = 'druk willekeurige toets... '
Z4 = LEN(KOM4)
Z5 = LEN(KOM5)
KOL4 = INT((79-Z4)/2)
KOL5 = INT((79-Z5)/2)
LENGTE = 3
@ 5,KOL4 SAY '&KOM4'
@ 6,KOL4
??REPLICATE('*',Z4)
@ 8,20 SAY 'De beschikbare opties:'
@10,20 SAY '[0]      Terug naar hoofdmenu'
@11,20 SAY '[1]      Backup van HD naar floppy'
@12,20 SAY '[2]      Restore van floppy naar HD'
KEUZE09 = -1
@ROW()+2,20 SAY 'Je keuze aub?      <0..2>'
@ROW(),60 GET KEUZE09 PICTURE "9" RANGE 0,2
READ
DO CASE
    CASE KEUZE09 = 0
        RETURN TO MASTER
    CASE KEUZE09 = 1
        DO EENANT WITH POST,DUMMY,KOM2,KOM3,LENGTE
        OPDR = IIF(READKEY()=12,'RETURN','')
        &OPDR
        IF SYSDRV = 'A'
            @ROW()+2,KOL5 SAY '&KOM5'
            SET ESCAPE OFF
            ?
            WAIT SPACE(KOL5)+'&KOM6' TO ANT
            SET ESCAPE ON
        ENDIF
        BBRONF = STUFF('&BRONF',LEN(BRONF),1,'*')
        RUN BACKUP &BBRONF &POST
    CASE KEUZE09 = 2
        DO EENANT WITH POST,DUMMY,KOM1,DUMMY,LENGTE
        OPDR = IIF(READKEY()=12,'RETURN','')
        &OPDR
        IF SYSDRV = 'A'
            @ROW()+2,KOL5 SAY '&KOM5'
            SET ESCAPE OFF
            ?
            WAIT SPACE(KOL5)+'&KOM6' TO ANT
            SET ESCAPE ON
        ENDIF
        CLOSE DATABASES
        BDOELF = STUFF('&DOELF',LEN(DOELF),1,'*')
        RUN RESTORE &POST &BDOELF

```

```

USE &DOELF
SET TALK ON
SET INDEX TO &NDX_1,&NDX_2,&NDX_3,&NDX_4
REINDEX
SET TALK OFF
CLOSE ALL
ENDCASE
RETURN

```

Deze module regelt de BACKUP en RESTORE operaties van het programma. Hierbij wordt zowel het standaardbestand, LIT.DBF, als het memobestand behandeld. Bij het uitvoeren van de RESTORE opdracht worden de indexen hersteld.

```

***** BDSCHP2.PRG
*
CLEAR
@10,20 SAY 'het hoogste recordnr is...' + LTRIM(STR(RECCOUNT()))
SET ESCAPE OFF
@12,0
WAIT SPACE(20) + 'druk willekeurige toets...' ' TO ANT
SET ESCAPE ON
RETURN

```

De gebruiker wordt in kennis gesteld van het laatste recordnr uit het gegevensbestand.

```

***** BLAD.PRG
*
PARAMETERS PAG,TEST,TESTWRD,NLN,PPRON,PPRUT
HOOFD = TEST + ': ' + TESTWRD
ZH = LEN(HOOFD)
IF ZH > 75
    HOOFD = LEFT(HOOFD,72) + '...'
    ZH = 75
ENDIF
ZK = LEN(TESTWRD)
LREC = .F.
DO WHILE .NOT. LREC
    TELIN = 0
    PAG = PAG + 1
    ?
    ?HOOFD + SPACE(80-ZH) + DTOC( DATE() ) + SPACE(12)
    ?? 'blz ' + LTRIM(STR(PAG)) + PPRUT
    ?
    TELIN = TELIN + 3
    DO WHILE TELIN < NLN-6
        DO DRUKREC
        TELIN = TELIN + 6
        SKIP
        IF (TESTWRD <> LEFT(&TEST,ZK)) .OR. EOF()
            LREC = .T.

```

```

        EXIT
    ENDIF
ENDDO
??PPRON
DO WHILE TELLN < NLN
    ?
    TELLN = TELLN + 1
ENDDO
ENDDO
RETURN

```

Deze module controleert de bladspiegel bij het uitvoeren van bepaalde afdrukopdrachten.

```

***** BLAD2.PRG
*
PARAMETERS PAG,HOOFD,NLN,PPRON,PPRUT
ZH = LEN(HOOFD)
DO WHILE .NOT. EOF()
    TELLN = 0
    PAG = PAG + 1
    ?
    ?HOOFD+SPACE(80-ZH)+DIOC( DATE() )+SPACE(12)
    ??'blz '+LTRIM(STR(PAG))+PPRUT
    ?
    TELLN = TELLN + 3
    DO WHILE TELLN < NLN-6 .AND. .NOT. EOF()
        DO DRUKREC
        TELLN = TELLN + 6
        SKIP
    ENDDO
    ??PPRON
    DO WHILE TELLN < NLN
        ?
        TELLN = TELLN + 1
    ENDDO
ENDDO
RETURN

```

Deze module controleert de bladspiegel bij het uitvoeren van bepaalde afdrukopdrachten.

```

***** BLAD3.PRG
*
PARAMETERS HOOFD,NLN
ZH = LEN(HOOFD)
TELLN = 0
?
?HOOFD+SPACE(80-ZH)+DIOC( DATE() )
?
TELLN = TELLN + 3

```

```

DO DRUKREC
TELLN = TELLN + 6
DO WHILE TELLN < NLN
?
    TELLN = TELLN + 1
ENDDO
RETURN

```

Deze module maakt de bladspiegel op bij het uitvoeren van
bepaalde afdrukopdrachten.

```

***** BLAD4.PRG
*
PARAMETERS PAG,NLN,TEST,TESTWRD,PPRON,PPRUT,L1,L2
HOOFD = 'recordnrs van '+STR(TEST,L1)+' tot '+STR(TESTWRD,L2)
ZH = LEN(HOOFD)
LREC = .F.
DO WHILE .NOT. LREC
    TELLN = 0
    PAG = PAG + 1
    ?
    ?HOOFD+SPACE(80-ZH)+DTC(DATE())+SPACE(12)
    ??'blz '+LTRIM(STR(PAG))+PPRUT
    ?
    TELLN = TELLN + 3
    DO WHILE TELLN < NLN-6
        GOTO RECORD TEST
        DO DRUKREC
        TELLN = TELLN + 6
        SKIP
        IF EOF() .OR. RECNO() > TESTWRD
            LREC = .T.
            EXIT
        ENDIF
    ENDDO
    ??PPRON
    DO WHILE TELLN < NLN
        ?
        TELLN = TELLN + 1
    ENDDO
ENDDO
RETURN

```

Dit onderdeel controleert de bladspiegel bij het uit-
voeren van bepaalde afdrukopdrachten.

```

***** BLAD5.PRG
*
PARAMETERS PAG,TEST,TESTWRD,NLN,PPRON,PPRUT
HOOFD = TEST+' ': '+TESTWRD
ZH = LEN(TRIM(HOOFD))
DO WHILE .NOT. EOF()

```

```

TELLN = 0
PAG = PAG + 1
?
?HOOFD+SPACE(80-ZH)+DIOC( DATE())+SPACE(12)+'blz '+LTRIM(STR(PAG))
??PPRUT
?
TELLN = TELLN+3
DO WHILE TELLN < NLN-6 .AND. .NOT. EOF()
    DO DRUKREC
    TELLN = TELLN+6
    CONTINUE
ENDDO
??PPRON
DO WHILE TELLN < NLN
    ?
    TELLN = TELLN+1
ENDDO
ENDDO
RETURN

```

Deze module controleert de bladspiegel bij het uitvoeren
van bepaalde afdrukopdrachten.

```

***** CAND.PRG
*
PARAMETER TEST,RANG,NLENGTE,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,;
        LYNNEN,PAPON,PAPUT,VLAG,KANTLIJN,MERK,CONDENS,POST,WRKF
CLEAR
KOM1 = 'volledige pathnaam doelbestand.DBF      (STOPPEN: ESC) '
KOM2 = 'F10 = vorige invoer'
STORE '' TO BESLUIT,DUMMY,DOELF,MAUTEUR,MTITEL,MDRAGER,MPLAATS,;
        MAARD,MTREFWRD,KRIT1
LENGTE = 60
DO WHILE .T.
    DO EENANT WITH DOELF,DUMMY,KOM1,KOM2,LENGTE
    IF READKEY() = 12
        EXIT
    ENDIF
    DUMMY = DOELF
    OPDR = IIF(LEFT('&DOELF',1)#POST,'LOOP','')
    &OPDR
    DO FEXIST WITH DOELF,BESLUIT
    OPDR = IIF(BESLUIT$'TtNnOo','','LOOP')
    &OPDR
    USE &BRONF
    DO OPDR604
    @12,20 SAY "werken met &TEST"
    SET FILTER TO FILE LIT
    LOCATE
    IF FOUND()
    DO CASE
        CASE BESLUIT$'Tt'
            SELECT 2
            USE &DOELF
            SELECT 1

```

```

DO WHILE .NOT. EOF()
  MAUTEUR = AUTEUR
  MTITEL = TITEL
  MDRAGER = DRAGER
  MPLAATS = PLAATS
  MAARD = AARD
  MTREFWRD = TREFWRD
  SELECT 2
  APPEND BLANK
  REPLACE AUTEUR WITH MAUTEUR,TITEL WITH MTITEL,;
    DRAGER WITH MDRAGER,PLAATS WITH MPLAATS,;
    AARD WITH MAARD,TREFWRD WITH MTREFWRD
  SELECT 1
  SKIP
ENDDO
CASE BESLUIT$'Nn'
  SELECT 2
  CREATE &DOELF FROM OPBOUW
  USE
  SELECT 1
  SET SAFETY OFF
  COPY TO &DOELF ALL
  SET SAFETY ON
CASE BESLUIT$'Oo'
  SET SAFETY OFF
  COPY TO &DOELF ALL
  SET SAFETY ON
ENDCASE
SET FILTER TO
CLOSE ALL
ELSE
  DO OPDR601 WITH KRIT1
ENDIF
ENDDO
RETURN

```

Deze module staat in voor het kopiëren van steekkaarten naar het scherm, de printer of een ander bestand op één van de beschikbare drives. De selectie van de te kopiëren steekkaarten gebeurt met behulp van de QUERY mogelijkheden van dBASEIII+.

```

***** CNR.PRG
*
PARAMETER TEST,RANG,NLENGTE,BRONF,BNDX 1,BNDX 2,BNDX 3,BNDX 4,;
  LYNEN,PAPON,PAPUT,VLAG,KANTLIJN,MERK,CONDENS,POST,WRKF
CLEAR
KOM1 = 'volledige pathnaam doelbestand.DBF'      (STOPPEN: ESC) '
KOM2 = 'F10 = vorige invoer'
KOM3 = 'hoogste recordnr in bestand...'
KOM4 = TEST+'...' (STOPPEN: ESC) '
STORE '' TO DOELF,DUMMY,DUM,NR,LNR,BESLUIT
LENGTE = 60
DO WHILE LEFT(DOELF,1) #POST
  DO EENANT WITH DOELF,DUMMY,KOM1,KOM2,LENGTE

```

```

OPDR = IIF(READKEY()=12, 'RETURN', '')
&OPDR
DUMMY = DOELF
ENDDO
DO FEXIST WITH DOELF, BESLUIT
OPDR = IIF(UPPER(BESLUIT)='S', 'RETURN', '')
&OPDR
USE &BRONF
LNR = STR(RECCOUNT())
DO WHILE .T.
    DO EENANT WITH NR, DUM, KOM4, KOM2, NLENGTE
    OPDR = IIF(READKEY()=12, 'EXIT', '')
    &OPDR
    DUM = NR
    IF INT(VAL(NR)) > INT(VAL(LNR))
        CLEAR
        KOM3 = KOM3+LNR
        @10,20 SAY "&KOM3"
        SET ESCAPE OFF
        @12,20
        WAIT SPACE(20)+'Druk willekeurige toets... ' TO ANT
        SET ESCAPE ON
    ELSE
        DO CASE
            CASE UPPER(BESLUIT) = 'T'
                SELECT 2
                USE &DOELF
            CASE UPPER(BESLUIT) = 'O'
                SELECT 2
                USE &DOELF
                DELETE ALL
                PACK
                BESLUIT = 'T'
            CASE UPPER(BESLUIT) = 'N'
                SELECT 2
                CREATE &DOELF FROM OPBOUW
                BESLUIT = 'T'
        ENDCASE
        SELECT 1
        GOTO VAL(NR)
        MAUTEUR = AUTEUR
        MTITEL = TITEL
        MDRAGER = DRAGER
        MPLAATS = PLAATS
        MAARD = AARD
        MTREFWRD = TREFWRD
        SELECT 2
        APPEND BLANK
        REPLACE AUTEUR WITH MAUTEUR, TITEL WITH MTITEL, ;
            DRAGER WITH MDRAGER, PLAATS WITH MPLAATS, ;
            AARD WITH MAARD, TREFWRD WITH MTREFWRD
    ENDIF
ENDDO
CLOSE ALL
RETURN

```

Deze module kopieert één steekkaart uit het bestand naar het scherm, de printer of een ander bestand op één van de beschikbare drives. De selectie gebeurt met behulp van het volgnummer van het record in het fysieke bestand.

***** CNRS.PRG

*

PARAMETERS TEST,RANG,NLENGTE,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,;
LYNEN,PAPON,PAPUT,VLAG,KANTLIJN,MERK,CONDENS,POST,WRKF

CLEAR

KOM1 = 'volledige pathnaam doelbestand.DBF' (STOPPEN: ESC)'

KOM2 = 'beginnen met recordnr...'

KOM3 = 'eindigen met recordnr...'

KOM4 = 'hoogste recordnr in bestand...'

KOM5 = 'F10 = vorige invoer'

LENGTE = 60

STORE 0 TO NR1,NR2

STORE '' TO DOELF,DUMMY,BESLUIT,MAUTEUR,MTITEL,MDRAGER,MPLAATS,;
MAARD,MTREFWRD

DO WHILE .T.

DO EENANT WITH DOELF,DUMMY,KOM1,KOM5,LENGTE

OPDR = IIF(READKEY()=12,'EXIT','')

&OPDR

DUMMY = DOELF

OPDR = IIF(LEFT('&DOELF',1)#POST,'LOOP','')

&OPDR

DO FEXIST WITH DOELF,BESLUIT

OPDR = IIF(UPPER(BESLUIT)='S','EXIT','')

&OPDR

USE &BRONF

DO TWEEANT WITH NR1,NR2,KOM2,KOM3,KOM4

GOTO RECORD NR1

DO CASE

CASE UPPER(BESLUIT) = 'T'

SELECT 2

USE &DOELF

CASE UPPER(BESLUIT) = 'O'

SELECT 2

USE &DOELF

DELETE ALL

PACK

CASE UPPER(BESLUIT) = 'N'

SELECT 2

CREATE &DOELF FROM OPBOUW

ENDCASE

SELECT 1

DO WHILE RECNO() <= NR2

MAUTEUR = AUTEUR

MTITEL = TTITEL

MDRAGER = DRAGER

MPLAATS = PLAATS

MAARD = AARD

```

MTREFWRD = TREFWRD
SELECT 2
APPEND BLANK
REPLACE AUTEUR WITH MAUTEUR,TITEL WITH MTITEL,;
        DRAGER WITH MDRAGER,PLAATS WITH MPLAATS,;
        AARD WITH MAARD,TREFWRD WITH MTREFWRD
SELECT 1
SKIP
ENDDO
CLOSE ALL
DOELF = ''
ENDDO
RETURN

```

Een konsekutieve serie van steekkaarten kopiëren naar het scherm, de printer of een ander bestand op één van de beschikbare drives. De selectie gebeurt met behulp van de volgnrs in het fysische bestand van het eerste en laatste te behandelen record.

```

***** COPIE.PRG
*
PARAMETER TEST,RANG,NLENGTE,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,;
        LYNEN,PAPON,PAPUT,VLAG,KANTLIJN,MERK,CONDENS,POST,WRKF
CLEAR
STORE ' ' TO DOELF,KRIT,BESLUIT,DUMMY,DUM
KOM1 = "volledige pathnaam doelbestand.DBF      (STOPPEN: ESC)"
KOM2 = "F10 = vorige invoer"
KOM3 =TEST+'...      (STOPPEN:ESC) '
STORE 60 TO LENGTE
DO WHILE .T.
    DO EENANT WITH DOELF,DUMMY,KOM1,KOM2,LENGTE
    OPDR = IIF(READKEY()=12,'EXIT','')
    &OPDR
    DUMMY = DOELF
    OPDR = IIF(LEFT("&DOELF",1)("&POST",'','LOOP'))
    &OPDR
    DO FEXIST WITH DOELF,BESLUIT
    OPDR = IIF(UPPER(BESLUIT)='S','EXIT','')
    &OPDR
    DO EENANT WITH KRIT,DUM,KOM3,KOM2,NLENGTE
    OPDR = IIF(READKEY()=12,'EXIT','')
    &OPDR
    DUM = KRIT
    ZK = LEN(KRIT)
    DO CASE
        CASE UPPER(BESLUIT)='T'
            USE &DOELF
            APPEND FROM &BRONF FOR KRIT = LEFT(&TEST,ZK)
        CASE UPPER(BESLUIT)='N'
            CREATE &DOELF FROM OPBOUW
            USE &DOELF
            APPEND FROM &BRONF FOR KRIT = LEFT(&TEST,ZK)
        CASE UPPER(BESLUIT)='O'
            USE &BRONF

```

```

SET INDEX TO &BNDX_1,&BNDX_2,&BNDX_3,&BNDX_4
SET ORDER TO RANG
SET SAFETY OFF
SEEK "&KRIT"
COPY TO &DOELF WHILE KRIT = LEFT (&TEST,ZK)
SET SAFETY ON
ENDCASE
CLOSE ALL
DOELF = ''
ENDDO
RETURN

```

Steekkaarten kopiëren uit het bestand naar het scherm, de printer of een ander bestand op één van de beschikbare drives. De selectie gebeurt met behulp van de AUTEUR, de DRAGER, de PLAATS of de AARD.

***** CTRF.PRG

*

```

PARAMETER TEST,RANG,NLENGTE,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,;
        LYNEN,PAPON,PAPUT,VLAG,KANTLIJN,MERK,CONDENS,POST,WRKF

```

CLEAR

```

KOM1 = 'volledige pathnaam doelbestand.DBF      (STOPPEN: ESC) '

```

```

KOM2 = 'F10 = vorige invoer'

```

```

KOM3 = TEST+'...      (STOPPEN: ESC) '

```

```

STORE '' TO DOELF,DUMMY,DUM,BESLUIT,KRIT

```

```

LENGTE = 60

```

```

DO WHILE .T.

```

```

    DO EENANT WITH DOELF,DUMMY,KOM1,KOM2,LENGTE

```

```

    OPDR = IIF(READKEY()=12,'EXIT','')

```

```

    &OPDR

```

```

    DUMMY = DOELF

```

```

    DO FEXIST WITH DOELF,BESLUIT

```

```

    OPDR = IIF(UPPER(BESLUIT)='S','EXIT','')

```

```

    &OPDR

```

```

    DO EENANT WITH KRIT,DUM,KOM3,KOM2,NLENGTE

```

```

    OPDR = IIF(READKEY()=12,'EXIT','')

```

```

    &OPDR

```

```

    DUM = KRIT

```

```

    DO CASE

```

```

        CASE UPPER(BESLUIT)='T'

```

```

            USE &DOELF

```

```

            APPEND FROM &BRONF FOR '&KRIT'$TREFWRD

```

```

        CASE UPPER(BESLUIT)='N'

```

```

            CREATE &DOELF FROM OPBOUW

```

```

            USE &DOELF

```

```

            APPEND FROM &BRONF FOR '&KRIT'$TREFWRD

```

```

        CASE UPPER(BESLUIT)='O'

```

```

            USE &BRONF

```

```

            SET SAFETY OFF

```

```

            COPY TO &DOELF FOR '&KRIT'$TREFWRD

```

```

            SET SAFETY ON

```

```

    ENDCASE

```

```

    CLOSE ALL

```

```

    DOELF = ''

```

```
ENDDO
RETURN
```

Dit programmaonderdeel staat in voor het kopiëren van steekkaarten naar het scherm, de printer of een ander bestand op één van de beschikbare drives. De selectie gebeurt op grond van één TREFWRD.

```
***** DRUKREC.PRG
*
PRIVATE Z
?AUTEUR
Z = LEN(AUTEUR)
?? REPLICATE(' ',80-Z)
?? 'Recnr: '+LTRIM(STR(RECNO()))
?TITEL
?DRAGER
?PLAATS
Z = LEN(PLAATS)
?? REPLICATE(' ',80-Z)
?? AARD
? TREFWRD
?
RETURN
```

Deze module organiseert het afdrukken van de inhoud van één steekkaart.

```
***** EENANT.PRG
*
PARAMETER V1,V2,C1,C2,N
PRIVATE KOL
CLEAR
KOL = INT((79-(LEN("&C2")))/2)
SET FUNCTION 10 TO "&V2;"
@ 6,0 SAY '&C1'
JN = ' '
V1 = SPACE(N)
DO WHILE .T.
    @2,KOL SAY '&C2'
    @ 8,0 GET V1 PICTURE '@!'
    READ
    OPDR = IIF(READKEY()=12,'EXIT','')
    &OPDR
    DO WHILE .T.
        OPDR = IIF(N > 80,"@11,0 CLEAR","@10,0 CLEAR")
        &OPDR
        SET ESCAPE OFF
        WAIT SPACE(10)+'Is het gegeven OK?      <J/N>      ' TO JN
        SET ESCAPE ON
        OPDR = IIF(AT("&JN","JN")#0,"EXIT","LOOP")
        &OPDR
```

```

ENDDO
OPDR = IIF("&JN" = "J", "EXIT", "LOOP")
&OPDR
ENDDO
STORE LTRIM(TRIM(V1)) TO V1
CLEAR
SET FUNCTION 10 TO "edit"
RETURN

```

De gebruiker moet één enkel gegeven, bv de AUTEUR of de DRAGER, aan het programma verstrekken.

```

***** FEXIST.PRG
*
PARAMETERS NFILE,ANT
IF .NOT. FILE("&NFILE")
  ANT = 'N'
ELSE
  CLEAR
  @14,20 SAY '&NFILE bestaat!'
  ANT = '*'
  DO WHILE .T.
    @15,0
    WAIT SPACE(20)+'O(verschrijven S(toppen T(oevoegen? ' ;
    TO ANT
    OPDR = IIF("&ANT" $"OoSStt", 'EXIT', 'LOOP')
    &OPDR
  ENDDO
ENDIF
RETURN

```

Deze module onderzoekt of een bestand met een gegeven naam bestaat. Is dit het geval, dan krijgt de gebruiker de mogelijkheid om het bestaande bestand te overschrijven met de nieuwe steekkaarten, om de nieuwe records aan het bestand toe te voegen of om de operatie af te breken.

```

***** FORM.PRG
*
PARAMETERS SYSDRV
CLEAR
KOM1 = 'Formateren van floppies'
KOM2 = 'Plaats MS-DOS diskette in A:'
KOM3 = 'druk willekeurige toets...'
Z1 = LEN(KOM1)
Z2 = LEN(KOM2)
KOL1 = INT((79-Z1)/2)
KOL2 = INT((79-Z2)/2)
KEUZE10 = -1
@ 5,KOL1 SAY '&KOM1'
@ 6,KOL1
??REPLICATE("*",Z1)

```

```

@ 8,20 SAY 'de beschikbare opties:'
@10,20 SAY '[0]      naar hoofdmenu'
@11,20 SAY '[1]      in de A: drive'
@12,20 SAY '[2]      in de B: drive'
@ROW()+3,20 SAY 'Je keuze aub?      <0..2>'
@ROW(),60 GET KEUZE10 PICTURE "9" RANGE 0,2
RR = ROW()
READ
DO CASE
  CASE KEUZE10 = 0
    RETURN TO MASTER
  CASE KEUZE10 = 1
    IF SYSDRV = 'A'
      @ RR+3,KOL2 SAY "&KOM2"
      SET ESCAPE OFF
      ?
      WAIT SPACE(KOL2)+"&KOM3" TO ANT
      SET ESCAPE ON
    ENDIF
    RUN FORMAT A:
  CASE KEUZE10 = 2
    IF SYSDRV = 'A'
      @ RR+3,KOL2 SAY "&KOM2"
      SET ESCAPE OFF
      ?
      WAIT SPACE(KOL2)+"&KOM3" TO ANT
      SET ESCAPE OFF
    ENDIF
    RUN FORMAT B:
ENDCASE
RETURN

```

Dit programmaonderdeel zorgt voor het formateren van diskettes. Dit kan zowel in de A: als de B: drive gebeuren. Indien het opstarten van de PC vanuit de A: drive gebeurde, dan zal het programma de gebruiker verzoeken de geschikte MS-DOS diskette in de A: drive te plaatsen, waarna het aantikken van een willekeurige toets het programma verder laat lopen.

```

***** FOUT1.PRG
*
PARAMETER FOUT
CLEAR
DO CASE
  CASE FOUT = 1
    @8,10 SAY 'Het gevraagde bestand werd niet gevonden.'
    @10,10 SAY 'Kontroleer de gebruikte diskette en remedieer.'
    OK = ''
    DO WHILE .T.
      @14,0
      WAIT SPACE(25)+'O(k, N(iet OK....      ' TO OK
      OPDR = IIF(AT("&OK", "oOnN")=0, "LOOP", "EXIT")
      &OPDR
    ENDDO

```

```

IF UPPER(OK) = 'O'
    RETRY
ELSE
    CLOSE ALL
    RETURN TO MASTER
ENDIF
CASE FOUT = 7
    CLEAR
    @8,20 SAY 'Er bestaat een bestand met de doelnaam'
    DO WHILE .T.
        @14,0
        WAIT SPACE(20)+'O(verschrijven; N(iet_overschrijven      ';
        TO ANT
        OPDR = IFF(ANT$"oOnN","EXIT","LOOP"
        &OPDR
    ENDDO
    IF ANT$"Nn"
        CLOSE ALL
        RETURN
    ENDIF
CASE FOUT = 56
    @8,10 SAY 'De diskette is vol.'
    @10,10 SAY 'Er wordt naar het HOOFDMENU terug gekeerd'
    SET ESCAPE OFF
    @14,0
    WAIT SPACE (10)+'druk willekeurige toets...      ' TO OK
    SET ESCAPE ON
    CLOSE ALL
    RETURN TO MASTER
CASE FOUT = 125
    @ 8,20 SAY 'KONTROLEER PRINTER'
    @14,0
    SET ESCAPE OFF
    WAIT SPACE(15)+'Druk willekeurige toets...      ' TO ANT
    SET ESCAPE ON
    SET PRINT ON
OTHERWISE
    @ 8,20 SAY 'NIET BEHANDELDE FOUT * GA NAAR HOOFDMENU'
    @10,20 SAY 'FOUTKODE: '
    ??FOUT
    @14,20
    ??MESSAGE()
    @18,0
    WAIT SPACE(20)+'Druk willekeurige toets...      ' TO ANT
    SET ESCAPE ON
    SET PRINT OFF
    CLOSE ALL
    RETURN TO MASTER
ENDCASE
RETURN

```

Dit is een foutbehandelingsprocedure. Men dient te bedenken dat fouten die, tijdens de uitvoering, op het DOS niveau tot uiting komen door een dBASEIII+ programma NIET kunnen verwerkt worden. Bij het uitvoeren van een module waarin de printer gebruikt wordt, kan het niet ON LINE zijn van het toestel ondervangen worden. Een tekort

aan papier gedurende het drukken zet de printer OFF LINE en stuurt een foutcode op het DOS niveau naar de PC. Het hervatten van de afdrukoperaties na het herstellen van de fout kan derhalve niet. Om zeker te zijn dat de steekkaart die afgedrukt wordt bij het detekteren van de fout nog volledig op het papier komt, worden de variabelen PAPON en PAPUT gebezigd. Ze sturen het aan- en uitzetten van de detector voor het papier.

***** IN.PRG

*

PARAMETER RANG,TEST,DOELF,NDX_1,NDX_2,NDX_3,NDX_4,N,KEUZE

PRIVATE Z

KOM3 = TEST+'... (exakt *** STOPPEN: ESC)'

KOM = 'Druk F10 om vorige invoer te herhalen'

STORE ' ' TO KOM4,KOM6,KRIT1

USE &DOELF

SET INDEX TO &NDX_1,&NDX_2,&NDX_3,&NDX_4

SET EXACT ON

CLEAR

DO WHILE .T.

OPDR = IIF(RANG=0,"SET ORDER TO","SET ORDER TO RANG")

&OPDR

KRITO = KRIT1

DO EENANT WITH KRIT1,KRITO,KOM3,KOM,N

OPDR = IIF(READKEY()=12,'EXIT','')

&OPDR

KONTROLE = .T.

DO CASE

CASE &TEST = AUTEUR

SEEK "&KRIT1"

CASE &TEST = TITEL

GO TOP

LOCATE FOR &TEST = "&KRIT1"

CASE &TEST = DRAGER

SEEK "&KRIT1"

ENDCASE

IF FOUND()

DO WHILE .NOT. EOF()

CLEAR

DO CASE

CASE RECNO() = 1

KOM5 = 'Eerste record in het fysisch bestand'

CASE RECNO() = RECCOUNT()

KOM5 = 'Laatste record in het fysisch bestand'

OTHERWISE

KOM5 = ''

ENDCASE

Z1 = LEN(KOM5)

Z2 = LEN(KOM4)

Z3 = LEN(KOM6)

KOL1 = INT((79-Z1)/2)

KOL2 = INT((79-Z2)/2)

KOL3 = INT((79-Z3)/2)

@ 2,KOL1 SAY "&KOM5"

@ 3,KOL2 SAY "&KOM4"

```

DO LSSCHRM
@19,KOL3 SAY "&KOM6"
@21,20 SAY 'A(nte, P(ost, N(oteren, E(xit'
DO WHILE .T.
    @22,0
    WAIT SPACE(20)+'Je keuze aub?      <A/P/N/E>      'TO ANT
    OPDR2 = IIF("&ANT"$"APNE","EXIT","LOOP")
    &OPDR2
ENDDO
DO CASE
CASE ANT = "E"
    KONTROLE = .F.
    EXIT
CASE ANT = "N"
    KONTROLE = .T.
    EXIT
CASE ANT = "P"
    KOM6 = ''
    SKIP
    IF EOF().OR. KRIT1 # &TEST
        SKIP -1
        KOM6 = 'Laatste record in logisch bestand'
    ENDIF
CASE ANT = "A"
    KOM6 = ''
    SKIP -1
    IF BOF()
        KOM6 = 'Eerste record in logisch bestand'
    ELSE
        IF KRIT1 # &TEST
            SKIP
            KOM6 = 'Eerste record in logisch bestand'
        ENDIF
    ENDIF
ENDCASE
ENDDO
ENDIF
DO WHILE KONTROLE
    SET ORDER TO 1
    STORE SPACE(30) TO MAUTEUR,MPLAATS
    STORE SPACE(120) TO MTITEL,MIREFWRD
    MDRAGER = SPACE(80)
    MAARD = ' '
    DO INSCHRM WITH KEUZE,MAUTEUR,MTITEL,MDRAGER,MPLAATS,MAARD, ;
        MIREFWRD,KRIT1
DO CASE
CASE READKEY()=270
    APPEND BLANK
    REPLACE AUTEUR WITH LTRIM(MAUTEUR);
    TITEL WITH LTRIM(MTITEL),DRAGER WITH LTRIM(MDRAGER);
    PLAATS WITH LTRIM(MPLAATS),AARD WITH MAARD;
    TREFWRD WITH LTRIM(MIREFWRD)
    KONTROLE = .F.
    CHANGE FIELDS MEMO
CASE READKEY() = 12
    KONTROLE = .F.
OTHERWISE
    LOOP

```

```

ENDCASE
ENDDO
ENDDO
CLOSE ALL
SET EXACT OFF
RETURN

```

Bij het invoeren van gegevens kan een controle gebeuren aan de hand van de AUTEUR, de DRAGER of de TITEL. Wordt hierbij geen record konform het criterium gevonden, dan kan de gebruiker de gegevens op het invoerscherm intoetsen. Wordt echter wel een steekkaart gevonden waarvan het sleutelveld overeenstemt met de testkonditie, dan verschijnt het gevonden record in het leesscherm. Dit kan het eerste record zijn van een reeks, bv alle steekkaarten die op een gegeven auteur betrekking hebben. De gebruiker kan nu de inhoud van de steekkaarten één voor één op het scherm halen en zo uitmaken of een gegeven al dan niet reeds in het bestand voorkomt. Is dit het geval dan kan de gebruiker overgaan tot het aanbieden van een ander criterium, een ander auteur bv. Betreft het weliswaar dezelfde auteur maar is het toch een andere referentie, dan kan de gebruiker het invoerscherm oproepen. Nadat de trefwoorden ingevuld en de gegevens opgeslagen zijn, komt men terecht in het invoerscherm van het memoveld. Een hulpscherm leert de gebruiker de verdere afhandeling van het record.

```

***** INFO.PRG
*
PARAMETER DELAY_1
TEL = 0
CLEAR
@ 5,25 SAY 'Programma voor beheer referenties'
@ 7,25 SAY 'Versie 1.31'
@ 8,25 SAY 'Oostende *** november 1990'
@ 9,25 SAY 'W. Deschacht'
@11,25 SAY 'Zorg voor een voldoende voorraad'
@12,25 SAY 'geformatteerde diskettes voor het'
@13,29 SAY 'uitvoeren van een backup'
@16,25 SAY '<CR> aantikken.....'
@ 3,15 TO 18,65
DO WHILE TEL <= DELAY_1
    TEL = TEL + 1
    OPDR = IIF(INKEY()=13,"EXIT","LOOP")
    &OPDR
ENDDO
RETURN

```

Een openingsscherm met enkele gegevens over het programma verschijnt. Aan te stippen valt dat het makkelijk is om voldoende geformatteerde diskettes ter beschikking te hebben, maar dat bij het ontbreken hiervan het programma niet moet verlaten worden. De TOOLBOX bevat de mogelijk-

heden om onder de controle van het programma diskettes te formatteren.

***** INITO.PRG

*

PARAMETERS STDF,ANDX,DNDX,PNDX,SNDX,TUS_F,WRKF,OPBF,NFILE, ;
NDX_1,NDX_2,NDX_3,NDX_4,EINDE,DELAY_1

CLEAR

KOM1 = 'Werken met...'

KOM2 = 'De volledige pathnaam bestand.DBF... (STOPPEN: ESC)'

Z1 = LEN(KOM1)

KOL1 = INT((79-Z1)/2)

KEUZE0 = -1

DUMMY = ''

LENGTE = 60

@6,KOL1 SAY '&KOM1'

@7,KOL1

?? REPLICATE ('*',Z1)

@9,20 SAY 'De beschikbare opties....'

@11,20 SAY '[0] het programma afbreken'

@12,20 SAY '[1] standaardbestand'

@13,20 SAY '[2] ander bestand'

@ROW()+2,20 SAY 'Je keuze aub? <0..2>'

@ROW(),60 GET KEUZE0 PICTURE "9" RANGE 0,2

READ

DO CASE

CASE KEUZE0 = 0

EINDE = .T.

CASE KEUZE0 = 1

NFILE = STDF

NDX_1 = ANDX

NDX_2 = DNDX

NDX_3 = PNDX

NDX_4 = SNDX

CASE KEUZE0 = 2

CLEAR

DIR

@ROW()+3,20 SAY 'Dit zijn de gebruikte bestandsnamen'

TEL = 0

DO WHILE TEL <= DELAY_1

TEL = TEL + 1

ENDDO

CLEAR

@2,20 SAY 'Dit zijn voorbehouden bestandsnamen...'

@4,30 SAY '&STDF'

@5,30 SAY '&TUS_F'

@6,30 SAY '&WRKF'

@7,30 SAY '&OPBF'

TEL = 0

DO WHILE TEL <= DELAY_1

TEL = TEL + 1

ENDDO

DO EENANT WITH NFILE,DUMMY,KOM2,DUMMY,LENGTE

IF READKEY()=12

EINDE = .T.

RETURN

```

ENDIF
OPDR = IIF(FILE('&NFILE'),'','CREATE &NFILE FROM &OPBF')
&OPDR
DO NDXEN WITH NFILE,NDX_1,NDX_2,NDX_3,NDX_4
IF .NOT. FILE('&NDX_1')
    INDEX ON AUTEUR TO &NDX_1
    INDEX ON DRAGER TO &NDX_2
    INDEX ON PLAATS TO &NDX_3
    INDEX ON AARD TO &NDX_4
ENDIF
ENDCASE
RETURN

```

Dit programmaonderdeel komt onmiddellijk na het openings-
 scherm. Hierin wordt de gebruiker gevraagd of er met het
 standaardbestand gewerkt wordt. Is dit niet het geval,
 dan toont dit onderdeel de bestanden die op de default-
 drive voorhanden zijn en vervolgens de voorbehouden na-
 men. Daarna kan de gebruiker de eigen gekozen benaming
 invoeren. De module controleert of dit bestand aanwezig
 is in de aangegeven subdirectory. Is dit niet het geval
 dan wordt het bestand gekreëerd. Vervolgens zorgt de mo-
 dule ervoor dat ook de indexbestanden onder een geschikte
 benaming worden aangemaakt. Het maken van het .DBF bestand
 houdt het creëren van een .DBT bestand in.

```

***** INSCHRM.PRG
*
PARAMETER KEUZE,MAUTEUR,MTITEL,MDRAGER,MPLAATS,MAARD,MTREFWRD, ;
        KRIT1
@ 0,0 CLEAR TO 22,79
DO CASE
    CASE KEUZE = 1
        MAUTEUR = KRIT1
    CASE KEUZE = 2
        MTITEL = KRIT1
    CASE KEUZE = 3
        MDRAGER = KRIT1
    OTHERWISE
ENDCASE
@ 2,29 SAY 'GEGEVENS INTOETSEN'
@ 1,27 TO 3,49 DOUBLE
@ 5,4 SAY 'Auteur      :'
@ 5,16 GET MAUTEUR FUNCTION "!"
@ 7,4 SAY 'Titel       :'
@ 7,16 GET MTITEL FUNCTION "!S60"
@ 9,4 SAY 'Drager      :'
@ 9,16 GET MDRAGER FUNCTION "!S60"
@ 11,4 SAY 'Plaats      :'
@ 11,16 GET MPLAATS FUNCTION "!"
@ 11,58 SAY 'Aard: '
@ 11,66 GET MAARD PICTURE "!"
@ 13,4 SAY 'Trefwoord  :'
@ 13,16 GET MTREFWRD FUNCTION "!S60"
@ 16,13 SAY "Cursor beweging      Wissen      Einde"

```

```

@18,17 SAY chr(024)+" HOME letter: Del Opslaan: ^END"
@19,15 SAY CHR(027)+SPACE(3)+CHR(026)
@19,21 SAY "END woord : ^T ^W"
@20,17 SAY CHR(025)+" veld : ^Y Afbreken: ESC"
@15,10 TO 22,70 DOUBLE
@17,11 TO 17,69
READ
RETURN

```

Deze module regelt het invoeren van de gegevens. Het bevat daartoe de nodige instructies.

```

***** INSTR.PRG
*
PARAMETER VLAG,PLN,KANTLLJN,MERK,CONDENS
CLEAR
@6,20 SAY 'Zorg voor voldoende kettingpapier...'
@7,20 SAY 'Zet printkop te paard op scheurlijn...'
@8,20 SAY 'Schakel stroom in...'
@9,20 SAY 'Zet printer ON LINE...'
SET ESCAPE OFF
@15,20 SAY '<H> = uitvoeren hoofdmenu...'
WAIT SPACE(20)+'andere toets = verder met uitvoering... ' TO ANT
SET ESCAPE ON
IF ANT $'Hh'
    RETURN TO MASTER
ENDIF
SET PRINT ON
SET MARGIN TO KANTLLJN
??CONDENS
??MERK
SET PRINT OFF
VLAG = .F.
CLEAR
ANDERS = IIF(PLN=72,66,72)
@6,10 SAY 'De standaardlengte bedraagt 66 of 72 lijnen'
@8,10 SAY 'De aktuele lengte belooft '+LTRIM(STR(PLN));
        +' lijnen'
ANT = '*'
DO WHILE .T.
    @10,20 SAY 'Indien OK druk <O>'
    @11,0
    WAIT SPACE(20)+'Anders <V> ' TO ANT
    OPDR = IIF(AT(ANT,"oovv")=0,"LOOP","EXIT")
    &OPDR
ENDDO
OPDR = IIF(UPPER(ANT) = 'O','PLN=PLN','PLN=ANDERS')
&OPDR
RETURN

```

Deze module bevat de nodige instructies voor het gebruik van de printer. Er kan gekozen worden tussen een papier

lengte van 72 of 66 lijnen.

```
***** INZNDR.PRG
*
PARAMETER DOELF,NDX_1,NDX_2,NDX_3,NDX_4,KEUZE
PRIVATE Z,KOL
USE &DOELF
SET INDEX TO &NDX_1,&NDX_2,&NDX_3,&NDX_4
KRIT1 = SPACE(60)
JN = '*'
DO WHILE .T.
    CLEAR
    KONTOLE = .T.
    STORE SPACE(30) TO MAUTEUR,MPLAATS
    STORE SPACE(120) TO MTITEL,MIREFWRD
    MDRAGER = SPACE(80)
    MAARD = ' '
    KOM = 'Geen controle'
    Z = LEN(KOM)
    KOL = INT((79-Z)/2)
    SET FUNCTION 10 TO "&KRIT1"
    @23,KOL SAY "&KOM"
    DO WHILE KONTOLE
        DO INSHRM WITH KEUZE,MAUTEUR,MTITEL,MDRAGER,MPLAATS,MAARD, ;
            MIREFWRD,KRIT1
        DO CASE
            CASE READKEY() = 270
                APPEND BLANK
                REPLACE AUTEUR WITH LTRIM(MAUTEUR) ;
                TITEL WITH LTRIM(MTITEL),DRAGER WITH LTRIM(MDRAGER) ;
                PLAATS WITH LTRIM(MPLAATS), AARD WITH MAARD;
                TREFWRD WITH LTRIM(MTREFWRD)
                KRIT1 = MDRAGER
                KONTOLE = .F.
            CHANGE FIELDS MEMO
            CASE READKEY() = 12
                KONTOLE = .F.
            OTHERWISE
                LOOP
        ENDCASE
    ENDDO
    DO WHILE UPPER(JN) # 'J' .AND. UPPER(JN) # 'N'
        JN = '*'
        @23,0
        WAIT SPACE(20) + 'Een steekkaart invoeren?      <j/n>      ' TO JN
    ENDDO
    IF UPPER(JN) = 'N'
        EXIT
    ENDIF
    JN = '*'
ENDDO
CLOSE ALL
SET FUNCTION 10 TO 'edit'
RETURN
```

Deze module regelt het invoeren van gegevens zonder een
kontrole op het reeds aanwezig zijn van de gegevens.

```
***** IO.PRG
*
PARAMETER KEUZE,RANG,N,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,LYNEN, ;
          PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,WRKF
PRIVATE Z,KOL
CLEAR
KOM3 = KEUZE+"...      (STOPPEN: ESC)"
KOM2 = 'F10 = vorige invoer'
GEW = .F.
STORE ' ' TO MAUTEUR,MITTEL,MDRAGER,MPLAATS,MAARD,MITREFWRD, ;
          KRIT1,KOM1,DUMMY
USE &NFILE
SET INDEX TO &NDX_1,&NDX_2,&NDX_3,&NDX_4
OPDR = IIF(RANG=0,"SET ORDER TO","SET ORDER TO RANG")
&OPDR
SELECT 2
USE &WRKF
DO WHILE .T.
  DO EENANT WITH KRIT1,DUMMY,KOM3,KOM2,N
  IF READKEY() = 12
    EXIT
  ELSE
    DUMMY = KRIT1
    KOM = KEUZE+'...      &KRIT1'
    Z = LEN(KOM)
    KOL = INT((79-Z)/2)
    @22,KOL SAY "&KOM"
    SELECT 1
    SEEK "&KRIT1"
    IF FOUND()
      Z3 = LEN("&KRIT1")
      DO WHILE KRIT1 = LEFT(UPPER(&KEUZE),Z3)
        REC = RECNO()
        SELECT 2
        APPEND BLANK
        REPLACE RECNR WITH REC
        SELECT 1
        SKIP
      ENDDO
    SELECT 2
    GO TOP
    REC = RECNR
    KOL1 = 0
    LEERSTE = .T.
    LLAATSTE = IIF(RECNO()=RECCOUNT(),.T.,.F.)
    STOPPEN = .F.
    DO WHILE .NOT. STOPPEN
      SELECT 1
      GOTO REC
      DO KOMR_1 WITH KOM1,KOL1,LEERSTE,LLAATSTE
```

```

C3 = IIF(DELETED(),'logisch geschrapt',' ')
Z3 = LEN(C3)
KOL3 = INT((79-Z3)/2)
C6 = IIF(GEW,'gewijzigd','')
GEW = .F.
@2,0 CLEAR TO 3,79
@2,KOL1 SAY "&KOM1"
@3,0 SAY "&C6"
@3,KOL3 SAY "&C3"
DO LSSCHRM
@21,10 SAY "A(nte P(ost W(ijzig S(chrap O(pvissen";
      +" M(emo E(xit"
ANT = '*'
DO WHILE .T.
  @21,70 GET ANT PICTURE "!"
  READ
  OPDR = IIF(AT("&ANT","APWSOME")=0,"LOOP","EXIT")
  &OPDR
ENDDO
DO WISSEL_1 WITH ANT,GEW,REC,STOPPEN,MAUTEUR,MTITEL, ;
      MDRAGER,MPLAATS,MAARD,MTREFWRD,RANG
LEERSTE = IIF(RECNO()=1,.T.,.F.)
LIAATSTE = IIF(RECNO()=RECCOUNT(),.T.,.F.)
DO LGSCHRM
ENDDO
SELECT 2
DELETE ALL
PACK
STOPPEN = .F.
GEW = .F.
ELSE
  DO OPDR601 WITH KRIT1
ENDIF
ENDIF
ENDIF
ENDDO
CLOSE ALL
RETURN

```

Deze module regelt het raadplegen van de steekkaarten op het scherm. De selectie gebeurt of via de auteur, de drager, de plaats of de aard. De gebruiker kan door alle records bladeren, die konform het criterium zijn, er kan logisch geschrapt worden of de schrapping kan ongedaan gemaakt worden en er kunnen in IEDER veld, inbegrepen het sleutelveld, veranderingen aangebracht worden.

***** IOAND.PRG

*

```

PARAMETER KEUZE,RANG,N,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,LYNEN, ;
      PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,WRF

```

CLEAR

C5 = 'zoeken via assist...'

Z5 = LEN('&C5')

KOL5 = INT((79-Z5)/2)

GEW = .F.

```

STORE ' ' TO MAUTEUR,MTITEL,MDRAGER,MPLAATS,MAARD,MIREFWRD, ;
      KOM1,KRIT1
USE &NFILE
SET INDEX TO &NDX_1,&NDX_2,&NDX_3,&NDX_4
SET ORDER TO
SELECT 2
USE &WRKF
DO WHILE .T.
  SELECT 1
  CLEAR
  DO OPDR604
  @ 10,KOL5 SAY '&C5'
  SET FILTER TO FILE LIT
  LOCATE
  IF FOUND()
    DO WHILE .NOT. EOF()
      REC = RECNO()
      SELECT 2
      APPEND BLANK
      REPLACE RECNR WITH REC
      SELECT 1
      SKIP
    ENDDO
  SET FILTER TO
  CLEAR
  SELECT 2
  GO TOP
  REC = RECNR
  KOL = 0
  LEERSTE = .T.
  STOPPEN = .F.
  LLAATSTE = IIF(RECNO()=RECCOUNT(),.T.,.F.)
  DO WHILE .NOT. STOPPEN
    SELECT 1
    GOTO REC
    DO KOMTR 1 WITH KOM1,KOL,LEERSTE,LLAATSTE
    C3 = IIF(DELETED(),'logisch geschrap','')
    Z3 = LEN('&C3')
    KOL3 = INT((79-Z3)/2)
    C6 = IIF(GEW,'GEWLJZIGD','')
    GEW = .F.
    @2,0 CLEAR TO 3,79
    @2,KOL SAY '&KOM1'
    @3,0 SAY '&C6'
    @3,KOL3 SAY '&C3'
    DO LSSCHRM
    @22,KOL5 SAY '&C5'
    @21,10 SAY;
    'A(nte P(ost W(ijzig S(chrap O(pvissen M(emo E(xit'
    ANT = '*'
  DO WHILE .T.
    @21,70 GET ANT PICTURE "!"
    READ
    OPDR = IIF(AT("&ANT","APWSOME")=0,"LOOP","EXIT")
    &OPDR
  ENDDO
  DO WISSEL_1 WITH ANT,GEW,REC,STOPPEN,MAUTEUR,MTITEL, ;
                MDRAGER,MPLAATS,MAARD,MIREFWRD,RANG

```

```

        LEERSTE = IIF(RECNO()=1,.T.,.F.)
        LLAATSTE = IIF(RECNO()=RECCOUNT(),.T.,.F.)
        DO LGSCHRM
        ENDDO
    ELSE
        DO OPDR601 WITH KRIT1
    ENDIF
    SELECT 2
    DELETE ALL
    PACK
    SELECT 1
    SET FILTER TO
    JN = '*'
    DO WHILE UPPER(JN) <> 'J' .AND. UPPER(JN) <> 'N'
        CLEAR
        @15,0
        WAIT SPACE(25)+'Nog een selectie?      <j/n>      ' TO JN
    ENDDO
    GEW = .F.
    IF UPPER (JN) = 'N'
        EXIT
    ENDIF
ENDDO
CLOSE ALL
RETURN

```

Deze module regelt het raadplegen van de steekkaarten op het scherm. De selectievoorwaarden worden samengesteld met behulp van de QUERY mogelijkheid van dBASEIII. De gebruiker kan bladeren door alle records die konform de selectiekriteria zijn. Er kan logisch geschrapt worden of de schrapping kan ongedaan gemaakt worden en er kunnen in IEDER veld veranderingen aangebracht worden.

```

***** IONR.PRG
*
PARAMETER KEUZE,RANG,N,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,LYNEN, ;
        PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,WRFK
CLEAR
STORE ' ' TO MAUTEUR,MITTEL,MDRAGER,MPLAATS,MAARD,MTREFWRD, ;
        KRIT1,DUMMY
C1 = KEUZE+'...      (STOPPEN: ESC) '
C3 = 'F10 = vorige invoer'
C5 = 'zoeken via EEN recordnr'
Z5 = LEN(C5)
KOL5 = INT((79-Z5)/2)
GEW = .F.
USE &NFILE
SET INDEX TO &NDX_1,&NDX_2,&NDX_3,&NDX_4
SET ORDER TO
DO WHILE .T.
    DO EENANT WITH KRIT1,DUMMY,C1,C3,N
    IF READKEY() = 12
        EXIT
    ENDIF

```

```

DUMMY = KRIT1
IF VAL("&KRIT1") > RECCOUNT()
  CLEAR
  @10,10 SAY 'Het hoogste recordnr is '+STR(RECCOUNT())
  @12,0
  SET ESCAPE OFF
  WAIT SPACE(20)+'Druk willekeurige toets...      ' TO ANT
  SET ESCAPE ON
  LOOP
ENDIF
GOTO RECORD VAL("&KRIT1")
DO WHILE .T.
  @0,0 CLEAR TO 20,79
  C3 = IIF(DELETED(),'logisch geschrapt','')
  Z3 = LEN(C3)
  KOL3 = INT((79-Z3)/2)
  KOM = IIF(GEW,'gewijzigd','')
  @2,0 CLEAR TO 3,79
  @2,KOL3 SAY "&C3"
  @3,0 SAY "&KOM"
  @22,KOL5 SAY '&C5'
  DO LSSCHRM
  @21,10 SAY;
  'W(ijzigen  S(schrappen  O(pvissen  M(emo  E(xit'
  ANT = '*'
  DO WHILE .T.
    @21,70 GET ANT PICTURE "!"
    READ
    OPDR = IIF(AT("&ANT","WSOME")=0,"LOOP","EXIT")
    &OPDR
  ENDDO
  DO WISSEL_3 WITH ANT,GEW,RECNO(),MAUTEUR,MITTEL,MDRAGER, ;
    MPLAATS,MAARD,MITREFWRD,RANG
  OPDR = IIF(ANT = "E","EXIT","LOOP")
  DO LGSCHRM
  &OPDR
ENDDO
GEW = .F.
ENDDO
CLOSE ALL
RETURN

```

Dit programmaonderdeel regelt het raadplegen van steekkaarten op het scherm. De selectie gebeurt via het volgnummer van het record. De gebruiker kan logisch schrappen of de schrapping ongedaan maken en in IEDER veld kunnen veranderingen aangebracht worden.

```

***** IONRS.PRG
*
PARAMETER KEUZE,RANG,N,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,LYNEN,;
  PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,WRKF
CLEAR
C1 = 'beginnen bij recordnr...'
C2 = 'eindigen met recordnr...'

```

```

C3 = 'laatste recordnr in bestand...'
STORE .F. TO STOPPEN,GEW
STORE 0 TO NR1,NR2,KOL2
STORE ' ' TO MAUTEUR,MTITEL,MDRAGER,MPLAATS,MAARD,MTREFWRD,;
      KOM2
Z5 = LEN(KEUZE)
KOL5 = INT((79-Z5)/2)
DO WHILE .NOT. STOPPEN
  USE &NFILE
  SET INDEX TO &NDX_1,&NDX_2,&NDX_3,&NDX_4
  SET ORDER TO
  DO TWEEANT WITH NR1,NR2,C1,C2,C3
  REC =NR1
  @22,KOL5 SAY '&KEUZE'
  LEERSTE = .T.
  LLAATSTE = IIF(REC=NR2,.T.,.F.)
  GOTO RECORD REC
  DO WHILE REC <= NR2
    C4 = IIF(DELETED()),'logisch geschrapd','')
    Z4 = LEN(C4)
    KOL4 = INT((79 - Z4)/2)
    KOM = IIF(GEW,'gewijzigd','')
    DO KOMTR_1 WITH KOM2,KOL2,LEERSTE,LLAATSTE
    @2,0 CLEAR TO 3,79
    @2,KOL2 SAY '&KOM2'
    @3,0 SAY '&KOM'
    @3,KOL4 SAY '&C4'
    DO LSSCHRM
    @21,10 SAY 'A(nte P(ost W(yzig S(chrap O(pvissen';
              +' M(emo E(xit'
    ANT = '*'
    DO WHILE .T.
      @21,70 GET ANT PICTURE "!"
      READ
      OPDR = IIF(AT("&ANT","APWSOME")=0,"LOOP","EXIT")
      &OPDR
    ENDDO
    DO WISSEL_2 WITH ANT,GEW,REC,RANG,MAUTEUR,MTITEL,MDRAGER,;
                  MPLAATS,MAARD,MTREFWRD,NR1,NR2
    LEERSTE = IIF(REC=NR1,.T.,.F.)
    LLAATSTE = IIF(REC=NR2,.T.,.F.)
    DO LGSCHRM
  ENDDO
  JN = '*'
  DO WHILE UPPER(JN) <> 'J' .AND. UPPER(JN) <> 'N'
    @22,79
    WAIT SPACE(23)+'Nog een selectie?      <j/n>      ' TO JN
  ENDDO
  IF UPPER(JN) = 'N'
    STOPPEN = .T.
  ENDIF
  GEW = .F.
ENDDO
CLOSE ALL
RETURN

```

Deze module organiseert het raadplegen op het scherm van een konsekutieve reeks steekkaarten. De selectie gebeurt met behulp van het volgnummer van het eerste en laatste te raadplegen record. De gebruiker kan door alle records bladeren, er kan logisch geschrapt worden of een schrapping kan ongedaan gemaakt worden en in IEDER veld kunnen veranderingen aangebracht worden.

***** IOT.PRG

*

PARAMETER KEUZE,RANG,N,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,LYNEN, ;
PAPON,PAPUT,VLAG,KANTLIJN,MERK,CONDENS,FLOPPY,WRKF

CLEAR

C1 = KEUZE+'... (STOPPEN: ESC) '

C2 = 'F10 = vorige invoer'

GEW = .F.

STORE ' ' TO MAUTEUR,MTITEL,MDRAGER,MPLAATS,MAARD,MTREFWRD, ;
DUMMY,KRIT

USE &NFILE

SET INDEX TO &NDX_1,&NDX_2,&NDX_3,&NDX_4

SET ORDER TO

SELECT 2

USE &WRKF

DO WHILE .T.

SELECT 1

DO EENANT WITH KRIT,DUMMY,C1,C2,N

IF READKEY() = 12

EXIT

ENDIF

DUMMY = KRIT

C5 = KEUZE+'... &KRIT'

Z5 = LEN("&C5")

KOL5 = INT((79-Z5)/2)

@22,KOL5 SAY "&C5"

GO TOP

LOCATE FOR "&KRIT"\$TREFWRD

IF FOUND()

DO WHILE .NOT. EOF()

REC = RECNO()

SELECT 2

APPEND BLANK

REPLACE RECNR WITH REC

SELECT 1

CONTINUE

ENDDO

SELECT 2

GO TOP

REC = RECNR

KOL1 = 0

LEERSTE = .T.

LLAATSTE = IIF(RECNO()=RECCOUNT(),.T.,.F.)

NIDRGN = .F.

KOM1 = ' '

```

DO WHILE .NOT. NTDRGN
  SELECT 1
  GOTO REC
  DO KOMTR_1 WITH KOM1,KOL1,LEERSTE,LLAATSTE
  C3 = IIF(DELETED(),'logisch geschrapt',' ')
  Z3 = LEN(C3)
  KOL3 = INT((79-Z3)/2)
  C6 = IIF(GEW,'gewijzigd','')
  GEW = .F.
  @2,0 CLEAR TO 3,79
  @2,KOL1 SAY "&KOM1"
  @3,0 SAY "&C6"
  @3,KOL3 SAY "&C3"
  DO LSSCHRM
  @21,10 SAY "A(nte P(ost W(ijzig S(chrap O(pvissen";
    +" M(emo E(xit"
  ANT = '*'
  DO WHILE .T.
    @21,70 GET ANT PICTURE "!"
    READ
    OPDR = IIF(AT("&ANT","APWSOME")=0,"LOOP","EXIT")
    &OPDR
  ENDDO
  DO WISSEL_1 WITH ANT,GEW,REC,NTDRGN,MAUTEUR,MTITEL,;
    MDRAGER,MPLAATS,MAARD,MTREFWRD,RANG
  LEERSTE = IIF(RECNO()=1,.T.,.F.)
  LLAATSTE = IIF(RECNO()=RECCOUNT(),.T.,.F.)
  DO LGSCHRM
ENDDO
SELECT 2
DELETE ALL
PACK
GEW = .F.
ELSE
  DO OPDR601 WITH KRIT
ENDIF
ENDDO
CLOSE ALL
RETURN

```

Dit programmaonderdeel regelt het raadplegen van steekkaarten op het scherm. De selectie gebeurt aan de hand van één trefwoord. De gebruiker kan snel bladeren door alle records, konform met het selectiekriterium, er kan logisch geschrapt worden of zo'n schrapping kan ongedaan gemaakt worden en er kunnen veranderingen aangebracht worden in IEDER veld.

```

***** KOMTR_1.PRG
*
PARAMETER TXT,SCHRMPST,X1,X2
DO CASE
  CASE X1 .AND. X2
    TXT = 'dit is een enig record'
  CASE X1

```

```

        TXT = 'dit is het eerste record'
CASE X2
        TXT = 'dit is het laatste record'
OTHERWISE
        TXT = ''
ENDCASE
SCHRMPST = INT((79-LEN(TXT))/2)
RETURN

```

De module bevat enkele kommentaarlijnen. In functie van de omstandigheden zal hiervan één lijn aan het programma doorgegeven worden.

```

***** LGSCHRM.PRG
*
@ 9,12
@10,12
@11,12
@12,12
@13,12
@14,12
@15,12 SAY SPACE(30)
@15,65 SAY ' '
@16,12
@17,12
@18,12
RETURN

```

Deze module zorgt voor het verwijderen van de gegevens uit het leesscherm.

```

***** LSSCHRM.PRG
*
POS = 0
POS1 = 0
@ 5, 31 SAY "Gevonden steekkaart"
@ 4, 29 TO 6,51 DOUBLE
@ 9, 0 SAY "Auteur      :"
@ 9, 12 SAY AUTEUR
@ 10, 0 SAY "Titel       :"
IF LEN(TRIM(TITEL)) <= 67
    @10, 12 SAY TITEL
ELSE
    MTITEL = TRIM(TITEL)
    LENGTE = LEN(MTITEL)
    DO SPLIT WITH POS,POS1,(MTITEL),(LENGTE)
    @ 10,12 SAY LEFT(MTITEL,POS)
    @ 11,12 SAY LTRIM(SUBSTR(MTITEL,POS+1,POS1))
    @ 12,12 SAY LTRIM(RIGHT(MTITEL,LENGTE-POS-POS1))
ENDIF
@ 13, 0 SAY "Drager      :"
IF LEN(TRIM(DRAGER)) <= 67

```

```

@ 13, 12 SAY DRAGER
ELSE
  MDRAGER = TRIM(DRAGER)
  LENGTE = LEN(MDRAGER)
  DO SPLIT WITH POS, POS1, (MDRAGER), (LENGTE)
  @ 13, 12 SAY LEFT(MDRAGER, POS)
  @ 14, 12 SAY LTRIM(RIGHT(MDRAGER, LENGTE-POS))
ENDIF
@ 15, 0 SAY "Plaats      :"
@ 15, 12 SAY PLAATS
@ 15, 55 SAY "Aard      :"
@ 15, 65 SAY AARD
@ 16, 0 SAY "Trefwoord  :"
IF LEN(TRIM(TREFWRD)) <= 67
  @ 16, 12 SAY TREFWRD
ELSE
  MTREFWRD = TRIM(TREFWRD)
  LENGTE = LEN(MTREFWRD)
  DO SPLIT WITH POS, POS1, (MTREFWRD), (LENGTE)
  @ 16, 12 SAY LEFT(MTREFWRD, POS)
  @ 17, 12 SAY LTRIM(SUBSTR(MTREFWRD, POS+1, POS1))
  @ 18, 12 SAY LTRIM(RIGHT(MTREFWRD, LENGTE-POS-POS1))
ENDIF
RETURN

```

De gegevens van een steekkaart worden door dit programmaonderdeel op een ordelijke wijze op het scherm gebracht.

```

***** LYST_TYD.PRG
*
PARAMETERS BRONF, BNDX, LYNEN, PAPON, PAPUT, MERK, KANTILLJN, ;
          CONDENS, VLAG
HOOFD = 'Lijst periodieken'
ZH = LEN(HOOFD)
KOLH = INT((79-ZH)/2)
BLZ = 0
DO INSTR WITH VLAG, LYNEN, KANTILLJN, MERK, CONDENS
CLEAR
@10, KOLH SAY '&HOOFD'
SET CONSOLE OFF
USE &BRONF
SET INDEX TO &BNDX
SET HEADING OFF
SET FILTER TO AARD = "A" .OR. AARD = "F"
SET DELETED ON
SET PRINT ON
GO TOP
POS = AT("(", DRAGER)
OPDR = IIF(POS > 0, "KRIT=LEFT(DRAGER, POS)", "KRIT=LEFT(DRAGER, 6)")
&OPDR
DO WHILE .NOT. EOF()
  TELLN = 0
  BLZ = BLZ + 1
  ?

```

```

?HOOFD
??REPLICATE(" ",80-ZH)
??DIOC(())
??REPLICATE(" ",100-88)
??'blz '+LTRIM(STR(BLZ))
??PAPUT
?
TELLN = TELLN + 3
DISPLAY DRAGER OFF TO PRINT
TELLN = TELLN+1
DO WHILE TELLN <= (LYNEN-6)
    SKIP
    OPDRO = IIF(EOF(),'EXIT','')
    &OPDRO
    POS = AT(" ",DRAGER)
    OPDR = IIF(POS=0,'POS=POS+6','')
    &OPDR
    OPDR1 = IIF(LEFT(DRAGER,POS)#KRIT','','LOOP')
    &OPDR1
    KRIT = LEFT(DRAGER,POS)
    DISPLAY DRAGER OFF TO PRINT
    TELLN = TELLN+1
ENDDO
??PAPON
DO WHILE TELLN < LYNEN
    ?
    TELLN = TELLN+1
ENDDO
ENDDO
SET PRINT OFF
SET CONSOLE ON
SET FILTER TO
SET HEADING ON
SET DELETED OFF
CLOSE ALL
RETURN TO MASTER

```

Deze module drukt de gehele inhoud van het veld DRAGER af. Voor ieder verschillende tijdschriftentitel gebeurt dit slechts éénmaal. Op deze wijze krijgt de gebruiker een lijst met voorbeelden en kan het aanwezig zijn van verschillende schrijfwijzen voor één welbepaald tijdschrift worden voorkomen.

```

***** MENU00.PRG
*
PARAMETERS TUS_F,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,VLAG,PIN, ;
            KANTLLJN,MERK,CONDENS,EINDE,WRKF,SYSDRV,PAPON,PAPUT
PRIVATE TITEL,Z,KOL
TITEL = 'H O O F D - M E N U'
Z = LEN('&TITEL')
KOL = INT((79-Z)/2)
FLOPPY = ''
DO WHILE .T.
    CLEAR

```

```

KEUZE00 = -1
@ 4,KOL SAY '&TITEL'
@ 5,KOL
?? REPLICATE('*',Z)
@ 7,20 SAY 'De beschikbare opties:'
@ 9,20 SAY '[0]   het programma verlaten'
@10,20 SAY '[1]   steekkaart(en) toevoegen'
@11,20 SAY '[2]   steekkaart(en) redigeren'
@12,20 SAY '[3]   steekkaart(en) wissen'
@13,20 SAY '[4]   steekkaart(en) kopiëren'
@14,20 SAY '[5]   andere opdracht'
@ROW()+3,20 SAY 'Je keuze aub?      <0..5>'
@ROW(),60 GET KEUZE00 PICTURE "9" RANGE 0,5
READ
DO CASE
  CASE KEUZE00 = 0
    EINDE = .T.
    EXIT
  CASE KEUZE00 = 1
    DO MENU01 WITH TUS_F,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,;
                  VLAG,PLN,KANTLIJN,MERK,CONDENS
  CASE KEUZE00 = 2
    TIT = 'STEEKKAART(EN) REDIGEREN'
    STORE 'IO' TO K1,K2,K3,K7
    K4 = 'IOT'
    K5 = 'IONR'
    K6 = 'IONRS'
    K8 = 'IOAND'
    DO MENU02 WITH TIT,K1,K2,K3,K4,K5,K6,K7,K8,FLOPPY,;
                  WRKF,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,;
                  PLN,PAPON,PAPUT,VLAG,KANTLIJN,MERK,CONDENS
  CASE KEUZE00 = 3
    DO MENU03 WITH NFILE,NDX_1,NDX_2,NDX_3,NDX_4,PLN,PAPON,;
                  PAPUT,VLAG,KANTLIJN,MERK,CONDENS,WRKF
  CASE KEUZE00 = 4
    DO MENU04 WITH NFILE,NDX_1,NDX_2,NDX_3,NDX_4,PLN,PAPON,;
                  PAPUT,VLAG,KANTLIJN,MERK,CONDENS,WRKF
  CASE KEUZE00 = 5
    DO MENU05 WITH NFILE,NDX_1,NDX_2,NDX_3,NDX_4,SYSDRV,PLN, ;
                  PAPON,PAPUT,MERK,KANTLIJN,CONDENS,VLAG
  OTHERWISE
ENDCASE
ENDDO
RETURN

```

Deze module bevat de primaire selecties, met name het invoeren, het wijzigen, het wissen en het kopiëren van steekkaarten alsmede het gebruik van een toolbox.

***** MENU01.PRG

*

PARAMETER TUS_F,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,VLAG, ;
 PLN,KANTLIJN,MERK,CONDENS

PRIVATE TITEL,Z,KOL

TITEL = 'STEEKKAART(EN) TOEVOEGEN'

```

Z = LEN('&TTITEL')
KOL = INT((79-Z)/2)
DO WHILE .T.
  CLEAR
  KEUZE01 = -1
  @ 4,KOL SAY '&TTITEL'
  @ 5,KOL
  ?? REPLICATE('*',Z)
  @ 7,20 SAY 'De beschikbare opties:'
  @ 9,20 SAY '[0] de module verlaten'
  @10,20 SAY '[1] toevoegen aan permanent bestand'
  @11,20 SAY '[2] toevoegen aan tijdelijk bestand'
  @12,20 SAY '[3] transfer van tijd. naar perm. bestand'
  @13,20 SAY '[4] schrappen tijdelijk bestand'
  @15,20 SAY 'Je keuze aub? <0..4>'
  @15,60 GET KEUZE01 PICTURE "9" RANGE 0,4
  READ
  DO CASE
    CASE KEUZE01 = 0
      EXIT
    CASE KEUZE01 = 1
      DO MENU06 WITH NFILE,NDX_1,NDX_2,NDX_3,NDX_4
    CASE KEUZE01 = 2
      DO MENU07 WITH (KEUZE01),TUS_F
    CASE KEUZE01 = 3
      DO MENU08 WITH TUS_F,NFILE,NDX_1,NDX_2,NDX_3,NDX_4, ;
        VLAG,PLN,KANTLLJN,MERK,CONDENS
    CASE KEUZE01 = 4
      DO MENU07 WITH (KEUZE01),TUS_F
  ENDCASE
ENDDO
RETURN

```

Het is mogelijk om de gegevens rechtstreeks aan het standaardbestand toe te voegen of om ze in een tijdelijk bestand op te slaan. De inhoud van dit tussentijdse bestand kan naar het standaardbestand overgeschreven worden en het tijdelijke bestand kan geschrapt worden.

***** MENU02.PRG

*

```

PARAMETER TTITEL,K1,K2,K3,K4,K5,K6,K7,K8,FLOPPY,WRKF,BRONF,BNDX_1, ;
  BNDX_2,BNDX_3,BNDX_4,LYNEN,PAPON,PAPUT,VLAG,KANTLLJN, ;
  MERK,CONDENS

```

```

PRIVATE TTITEL,Z,KOL

```

```

Z = LEN('&TTITEL')

```

```

KOL = INT((79-Z)/2)

```

```

DO WHILE .T.

```

```

  CLEAR

```

```

  KEUZE02 = -1

```

```

  @ 4,KOL SAY '&TTITEL'

```

```

  @ 5,KOL

```

```

  ?? REPLICATE('*',Z)

```

```

  @ 7,20 SAY 'De beschikbare mogelijkheden:'

```

```

  @ 9,20 SAY '[0] de module verlaten'

```

```

@10,20 SAY '[1] zoeken via auteur'
@11,20 SAY '[2] zoeken via drager'
@12,20 SAY '[3] zoeken via plaats'
@13,20 SAY '[4] zoeken via EEN trefwoord'
@14,20 SAY '[5] zoeken via EEN recordnummer'
@15,20 SAY '[6] zoeken via een interval van recordnrs'
@16,20 SAY '[7] zoeken via aard'
@17,20 SAY '[8] anders zoeken'
@19,20 SAY 'Je keuze aub? <0..8>'
@19,60 GET KEUZE02 PICTURE "9" RANGE 0,8
READ
DO CASE
CASE KEUZE02 = 0
EXIT
CASE KEUZE02 = 1
TESTVELD = 'auteur'
RANG = 1
N = 30
DO &K1 WITH TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3, ;
BNDX_4,LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK, ;
CONDENS,FLOPPY,WRKF
CASE KEUZE02 = 2
TESTVELD = 'drager'
RANG = 2
N = 80
DO &K2 WITH TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3, ;
BNDX_4,LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK, ;
CONDENS,FLOPPY,WRKF
CASE KEUZE02 = 3
TESTVELD = 'plaats'
RANG = 3
N = 30
DO &K3 WITH TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3, ;
BNDX_4,LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK, ;
CONDENS,FLOPPY,WRKF
CASE KEUZE02 = 4
TESTVELD = 'trefwoord'
RANG = 0
N = 30
DO &K4 WITH TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3, ;
BNDX_4,LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK, ;
CONDENS,FLOPPY,WRKF
CASE KEUZE02 = 5
TESTVELD = 'recordnr'
RANG = 0
N = 5
DO &K5 WITH TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3, ;
BNDX_4,LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK, ;
CONDENS,FLOPPY,WRKF
CASE KEUZE02 = 6
TESTVELD = 'een interval van recordnrs'
RANG = 0
N = 0
DO &K6 WITH TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3, ;
BNDX_4,LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK, ;
CONDENS,FLOPPY,WRKF
CASE KEUZE02 = 7
TESTVELD = 'AARD'

```

```

RANG = 4
N = 2
DO &K7 WITH TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3, ;
      BNDX_4,LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK, ;
      CONDENS,FLOPPY,WRKF
CASE KEUZE02 = 8
  TESTVELD = 'anders geformuleerde keuze'
  RANG = 0
  N = 0
  DO &K8 WITH TESTVELD,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3, ;
        BNDX_4,LYNEN,PAPON,PAPUT,VLAG,KANTLLJN,MERK, ;
        CONDENS,FLOPPY,WRKF
ENDCASE
ENDDO
RETURN

```

De gebruiker kan grasduinen doorheen het bestand. Hij kan daarbij verschillende wegen bewandelen door gebruik te maken van een geschikt selectiekriterium.

```

***** MENU03.PRG
*
PARAMETER BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,LYNEN,PAPON,PAPUT, ;
      VLAG,KANTLLJN,MERK,CONDENS,WRKF
PRIVATE TITEL,Z,KOL
TITEL = 'STEEKKAART(EN) WISSEN'
Z = LEN('&TITEL')
KOL = INT((79-Z)/2)
FLOPPY = ''
DO WHILE .T.
  CLEAR
  KEUZE03 = -1
  @ 6,KOL SAY '&TITEL'
  @ 7,KOL
  ?? REPLICATE('*',Z)
  @ 9,20 SAY 'De beschikbare opties:'
  @11,20 SAY '[0] de module verlaten'
  @12,20 SAY '[1] steekkaart(en) logisch wissen'
  @13,20 SAY '[2] controle gewiste steekkaart(en)'
  @14,20 SAY '[3] steekkaart(en) fysisch wissen'
  @17,20 SAY 'Je keuze aub? <0..3>'
  @17,60 GET KEUZE03 PICTURE "9" RANGE 0,3
  READ
  DO CASE
    CASE KEUZE03 = 0
      EXIT
    CASE KEUZE03 = 1
      STORE 'WIS' TO K1,K2,K3,K7
      K4 = 'WISTRF'
      K5 = 'WISNR'
      K6 = 'WISNRS'
      K8 = 'WISAND'
      DO MENU02 WITH TITEL,K1,K2,K3,K4,K5,K6,K7,K8,FLOPPY,WRKF, ;
            BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,LYNEN, ;
            PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS

```

```

CASE KEUZE03 = 2
  USE &BRONF
  GO TOP
  LOCATE FOR DELETED()
  DO WHILE .NOT. EOF()
    DO U_SCHRP
    CONTINUE
  ENDDO
  CLOSE ALL
CASE KEUZE03 = 3
  CLEAR
  @ 10,10 SAY "de gemerkte records worden verwijderd"
  @ 12,12 SAY "dit kan enige tijd in beslag nemen"
  USE &BRONF
  @ 14,0
  SET TALK ON
  PACK
  SET INDEX TO &BNDX_1,&BNDX_2,&BNDX_3,&BNDX_4
  REINDEX
  SET TALK OFF
  CLOSE ALL
ENDCASE
ENDDO
RETURN

```

Bij het wissen van steekkaarten is het mogelijk om logisch te wissen, een controle uit te voeren op de gemerkte records en zonodig het merkteken te verwijderen. Tenslotte kan men de logische geschrapte records ook fysisch uit het bestand verwijderen. De gebruiker moet voor ogen houden dat hierdoor NIETS geschrapt wordt uit het memobestand!

```

***** MENU04.PRG
*
PARAMETERS BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,LYNEN,PAPON,;
          PAPUT,VLAG,KANTLIJN,MERK,CONDENS,WRKF
PRIVATE TITEL,Z,KOL
TITEL = 'STEEKKAART(EN) KOPIEREN'
Z = LEN('&TITEL')
KOL = INT((79-Z)/2)
DO WHILE .T.
  CLEAR
  KEUZE04 = -1
  @ 4,KOL SAY '&TITEL'
  @ 5,KOL
  ?? REPLICATE ('*',Z)
  @ 7,20 SAY 'De beschikbare opties:'
  @ 9,20 SAY '[0] de module verlaten'
  @10,20 SAY '[1] kopie naar scherm'
  @11,20 SAY '[2] kopie naar printer'
  @12,20 SAY '[3] kopie naar floppy in A:'
  @13,20 SAY '[4] kopie naar floppy in B:'
  @14,20 SAY '[5] kopie op HDISK (C:)'
  @15,20 SAY '[6] kopie op HDISK (D:)'

```

```

@18,20 SAY 'Je keuze aub?      <0..6>'
@18,60 GET KEUZE04 PICTURE "9" RANGE 0,6
READ
DO CASE
    CASE KEUZE04 = 3
        POST = 'A'
    CASE KEUZE04 = 4
        POST = 'B'
    CASE KEUZE04 = 5
        POST = 'C'
    CASE KEUZE04 = 6
        POST = 'D'
    OTHERWISE
        POST = ''
ENDCASE
DO CASE
    CASE KEUZE04 = 0
        EXIT
    CASE KEUZE04 = 1
        TTL = 'RECORDS NAAR SCHERM SCHRIJVEN'
        K1 = 'IO'
        K2 = 'IO'
        K3 = 'IO'
        K4 = 'IOT'
        K5 = 'IONR'
        K6 = 'IONRS'
        K7 = 'IO'
        K8 = 'IOAND'
    CASE KEUZE04 = 2
        TTL = 'RECORDS AFDrukKEN'
        K1 = 'PRT'
        K2 = 'PRT'
        K3 = 'PRT'
        K4 = 'PTR'
        K5 = 'PNR'
        K6 = 'PNRS'
        K7 = 'PRT'
        K8 = 'PAND'
    CASE KEUZE04 >= 3
        TTL = 'RECORDS KOPIEREN OP SCHIJF IN &POST:'
        K1 = 'COPIE'
        K2 = 'COPIE'
        K3 = 'COPIE'
        K4 = 'CTRF'
        K5 = 'CNR'
        K6 = 'CNRS'
        K7 = 'COPIE'
        K8 = 'CAND'
ENDCASE
DO MENU02 WITH TTL,K1,K2,K3,K4,K5,K6,K7,K8,POST,WRKF,BRONF, ;
                BNDX_1,BNDX_2,BNDX_3,BNDX_4,LYNEN,PAPON,PAPUT, ;
                VLAG,KANTLIJN,MERK,CONDENS
ENDDO
RETURN

```

Overeenkomstig de wens van de gebruiker stuurt het pro-

gramma de gegevens naar het scherm, de printer of de gekozen diskdrive.

***** MENU05.PRG

*

PARAMETERS NFILE,NDX_1,NDX_2,NDX_3,NDX_4,SYSDRV,PLN,PAPON, ;

PAPUT,MERK,KANTLLJN,CONDENS,VLAG

PRIVATE TTITEL,Z,KOL

TTITEL = 'TOOLBOX'

Z = LEN('&TTITEL')

KOL = INT((79-Z)/2)

DO WHILE .T.

CLEAR

KEUZE05 = -1

@ 5,KOL SAY '&TTITEL'

@ 6,KOL

?? REPLICATE('*',Z)

@ 8,20 SAY 'De beschikbare opties:'

@10,20 SAY '[0] de module verlaten'

@11,20 SAY '[1] backup uitvoeren'

@12,20 SAY '[2] lijst periodieken afdrukken'

@13,20 SAY '[3] formateren floppies'

@ROW()+3,20 SAY 'Je keuze aub? <0..3>'

@ROW(),60 GET KEUZE05 PICTURE "9" RANGE 0,3

READ

DO CASE

CASE KEUZE05 = 0

EXIT

CASE KEUZE05 = 1

DO BCKP WITH NFILE,SYSDRV,NDX_1,NDX_2,NDX_3,NDX_4

CASE KEUZE05 = 2

DO LYST_TYD WITH NFILE,NDX_2,PLN,PAPON,PAPUT,MERK,;

KANTLLJN,CONDENS,VLAG

CASE KEUZE05 = 3

DO FORM WITH SYSDRV

ENDCASE

ENDDO

RETURN

Dit onderdeel verleent toegang tot de TOOLBOX mogelijkheden. Er kan een BACKUP of een RESTORE uitgevoerd worden, er kan een lijst met periodieken worden afgedrukt en er kunnen floppies worden geformateerd.

***** MENU06.PRG

*

PARAMETER DOELF,NDX_1,NDX_2,NDX_3,NDX_4

PRIVATE TTITEL,Z,KOL

TTITEL = 'KONTROLE BLJ INVOER'

Z = LEN('&TTITEL')

KOL = INT((79-Z)/2)

DO WHILE .T.

CLEAR

```

KEUZE06 = -1
@ 4,KOL SAY '&TITEL'
@ 5,KOL
?? REPLICATE('*',Z)
@ 7,20 SAY 'De beschikbare opties:'
@ 9,20 SAY '[0]   de module verlaten'
@10,20 SAY '[1]   middels de auteur'
@11,20 SAY '[2]   middels de titel'
@12,20 SAY '[3]   middels de drager'
@13,20 SAY '[4]   geen'
@17,20 SAY 'Je keuze aub?      <0..4>'
@17,60 GET KEUZE06 PICTURE "9" RANGE 0,4
READ
DO CASE
  CASE KEUZE06 = 0
    EXIT
  CASE KEUZE06 = 1
    RANG = 1
    TEST = 'auteur'
    N = 30
    DO IN WITH RANG,TEST,DOELF,NDX_1,NDX_2,NDX_3,NDX_4,N, ;
      KEUZE06
  CASE KEUZE06 = 2
    RANG = 0
    TEST = 'titel'
    N = 120
    DO IN WITH RANG,TEST,DOELF,NDX_1,NDX_2,NDX_3,NDX_4,N, ;
      KEUZE06
  CASE KEUZE06 = 3
    RANG = 2
    TEST = 'drager'
    N = 80
    DO IN WITH RANG,TEST,DOELF,NDX_1,NDX_2,NDX_3,NDX_4,N, ;
      KEUZE06
  CASE KEUZE06 = 4
    DO INZNR WITH DOELF,NDX_1,NDX_2,NDX_3,NDX_4,KEUZE06
ENDCASE
ENDDO
RETURN

```

De gegevens kunnen ingevoerd worden met of zonder een controle op het reeds voorhanden zijn. Wordt voor een controle gekozen, dan kan dit geschieden aan de hand van de inhoud van het veld AUTEUR, het veld TITEL of het veld DRAGER.

***** MENU07.PRG

*

PARAMETER NKEUZE,TUS_F

PRIVATE TITEL,Z,KOL

DO CASE

CASE NKEUZE = 2

TITEL = 'STEEKKAART(EN) TOEVOEGEN AAN HULPBESTAND'

CASE NKEUZE = 4

TITEL = 'TIJDELIJK BESTAND SCHRAPPEN'

```

ENDCASE
Z = LEN('&TITEL')
KOL = INT((79-Z)/2)
KEUZE07 = -1
CLEAR
DO WHILE .T.
    @ 4,KOL SAY '&TITEL'
    @ 5,KOL
    ?? REPLICATE('*',Z)
    @ 7,20 SAY 'Localisatie van het hulpbestand:'
    @ 9,20 SAY '[0] de module verlaten'
    @10,20 SAY '[1] hulpbestand op A:'
    @11,20 SAY '[2] hulpbestand op B:'
    @12,20 SAY '[3] hulpbestand op C:'
    @13,20 SAY '[4] hulpbestand op D:'
    @17,20 SAY 'Je keuze aub? <0..4>'
    @17,60 GET KEUZE07 PICTURE "9" RANGE 0,4
    READ
    DO CASE
        CASE KEUZE07 = 0
            EXIT
        CASE KEUZE07 = 1
            FLOPPY = "A:"
        CASE KEUZE07 = 2
            FLOPPY = "B:"
        CASE KEUZE07 = 3
            FLOPPY = "C:"
        CASE KEUZE07 = 4
            FLOPPY = "D:"
    ENDCASE
    DO TANK WITH FLOPPY,NKEUZE,TUS_F
ENDDO
RETURN

```

Indien bij het inschrijven van de gegevens een hulpbestand gebruikt wordt, of werd, dan dient de gebruiker aan te geven in welke drive het medium zich bevind waarop dit tijdelijke bestand moet gekreëerd of geschrapt worden.

```

***** MENU08.PRG
*
PARAMETER TUS_F,NFILE,NDX_1,NDX_2,NDX_3,NDX_4,VIAG,PLN, ;
    KANTLIJN,MERK,CONDENS
PRIVATE TITEL,Z,KOL
TITEL = 'TRANSFER STEEKKAART(EN) NAAR PERM. BESTAND'
Z = LEN('&TITEL')
KOL = INT((79-Z)/2)
DO WHILE .T.
    CLEAR
    KEUZE08 = -1
    @ 4,KOL SAY '&TITEL'
    @ 5,KOL
    ?? REPLICATE('*',Z)
    @ 7,20 SAY 'de beschikbare opties:'
    @ 9,20 SAY '[0] de module verlaten'

```

```

@10,20 SAY '[1]   transfer van A: -> &NFILE'
@11,20 SAY '[2]   transfer van B: -> &NFILE'
@12,20 SAY '[3]   transfer van C: -> &NFILE'
@13,20 SAY '[4]   transfer van D: -> &NFILE'
@17,20 SAY 'Je keuze aub?      <0..4>'
@17,60 GET KEUZE08 PICTURE "9" RANGE 0,4
READ
DO CASE
    CASE KEUZE08 = 0
        EXIT
    CASE KEUZE08 = 1
        FLOPPY = "A:"
    CASE KEUZE08 = 2
        FLOPPY = "B:"
    CASE KEUZE08 = 3
        FLOPPY = "C:"
    CASE KEUZE08 = 4
        FLOPPY = "D:"
ENDCASE
DO TRANSFER WITH (TUS_F),NFILE,NDX_1,NDX_2,NDX_3,NDX_4, ;
                VLAG,PLN,KANTLLJN,MERK,CONDENS,FLOPPY
ENDDO
RETURN

```

Werden de ingevoerde gegevens opgeslagen in een tijdelijk bestand, dan dient de gebruiker het programma te laten weten in welke drive het medium te vinden is waarop dit voorkomt zodat het programma de gegevens kan vinden om ze naar het permanente bestand over te schrijven.

```

***** NDXEN.PRG
*
PARAMETER AFILE,NDX_1,NDX_2,NDX_3,NDX_4
POS = AT(".", "&AFILE")
OPDR = IIF(POS=0, "STAM=AFILE", "STAM=LEFT(' &AFILE', POS-1)")
&OPDR
LS = LEN(STAM)
POS = AT(":", "&STAM")
FLOP = LEFT("&STAM", POS)
STAM = RIGHT("&STAM", LS-POS)
POS = 1
DO WHILE POS > 0
    LS = LEN(STAM)
    POS = AT("\", "&STAM")
    FLOP = FLOP+LEFT("&STAM", POS)
    STAM = RIGHT("&STAM", LS-POS)
ENDDO
NDX_1 = FLOP+"A"+STAM+".NDX"
NDX_2 = FLOP+"D"+STAM+".NDX"
NDX_3 = FLOP+"P"+STAM+".NDX"
NDX_4 = FLOP+"S"+STAM+".NDX"
RETURN

```

De namen van de indexbestanden worden overeenkomstig de naam van het gegevensbestand aangemaakt.

***** OPDR601.PRG

*

PARAMETER KRIT1

CLEAR

@ 6,0 SAY 'Het gegeven: '

@ 8,0 SAY "&KRIT1"

@10,0 SAY 'werd NIET gevonden!'

@14,0

SET ESCAPE OFF

WAIT SPACE(20)+'Druk willekeurige toets... ' TO ANT

SET ESCAPE ON

RETURN

Deze module deelt de gebruiker mede dat in het bestand GEEN steekkaart beantwoordt aan het gebruikte criterium.

***** OPDR604.PRG

*

CLEAR

?CHR(7)+CHR(7)

@6,0

TEXT

Zo direct verschijnt een hulpscherm
waarmede de selectievoorwaarden kunnen
vastgelegd worden.

Onthou:

lijn wissen: lijnnr kiezen <^U>

lijn invoegen: lijnnr kiezen <^N>

gebruiken: SAVE

ENDTEXT

@15,0

SET ESCAPE OFF

WAIT SPACE(30)+'Willekeurige toets drukken... ' TO ANT

MODIFY QUERY LIT.QRY

RETURN

Dit hulpscherm verschaft de gebruiker enkele elementaire richtlijnen voor het aanwenden van het QUERY scherm van dBASEIII+.

***** PAND.PRG

*

PARAMETERS KEUZE,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,LYNEN,;

```

        PAPON, PAPUT, VLAG, KANTLLJN, MERK, CONDENS, FLOPPY, WRKF
CLEAR
STORE '' TO KRIT
KOM3 = 'er wordt gezocht via ASSIST...'
IF VLAG
    DO INSTR WITH VLAG, LYNEN, KANTLLJN, MERK, CONDENS
ENDIF
USE &BRONF
DO WHILE .T.
    CLEAR
    DO OPDR604
    @10,20 SAY '&KOM3'
    SET FILTER TO FILE LIT
    LOCATE
    IF FOUND()
        BLZ = 0
        SET CONSOLE OFF
        SET PRINT ON
        DO BLAD2 WITH BLZ, KOM3, LYNEN, PAPON, PAPUT
        SET PRINT OFF
        SET CONSOLE ON
    ELSE
        DO OPDR601 WITH KRIT
    ENDIF
    SET FILTER TO
    JN = '*'
    CLEAR
    DO WHILE UPPER(JN) <> 'J' .AND. UPPER(JN) <> 'N'
        @15,0
        WAIT SPACE(25)+'Nog een selectie?      <j/n>      ' TO JN
    ENDDO
    OPDR = IIF(UPPER(JN)='N', 'EXIT', '')
    &OPDR
ENDDO
CLOSE ALL
RETURN

```

Deze module regelt het afdrukken van de steekkaarten die
geslecteerd worden met behulp van het QUERY scherm van
dBASEIII+

```

***** PNR.PRG
*
PARAMETER KEUZE, RANG, N, BRONF, BNDX_1, BNDX_2, BNDX_3, BNDX_4, LYNEN, ;
        PAPON, PAPUT, VLAG, KANTLLJN, MERK, CONDENS, FLOPPY, WRKF
CLEAR
STORE '' TO KRIT, DUMMY
C1 = KEUZE + '...      (STOPPEN: ESC) '
C2 = 'F10 = vorige invoer'
C3 = 'zoeken via...' + KEUZE
IF VLAG
    DO INSTR WITH VLAG, LYNEN, KANTLLJN, MERK, CONDENS
ENDIF
USE &BRONF
DO WHILE .T.

```

```

CLEAR
DO EENANT WITH KRIT,DUMMY,C1,C2,N
OPDR = IIF(READKEY( )=12, 'EXIT', '')
&OPDR
IF VAL("&KRIT") > RECCOUNT()
    DO BDSCHP2
ELSE
    GOTO RECORD VAL("&KRIT")
    SET CONSOLE OFF
    SET PRINT ON
    DO BLAD3 WITH C3,LYNEN
    SET PRINT OFF
    SET CONSOLE ON
ENDIF
ENDDO
CLOSE ALL
RETURN

```

Deze module draagt zorg voor het afdrukken van het record dat op grond van het volgnr in het fysische bestand werd geselecteerd.

```

***** PNRS.PRG
*
PARAMETER KEUZE,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,LYNEN,;
        PAPON,PAPUT,VLAG,KANTLIJN,MERK,CONDENS,FLOPPY,WRKF
CLEAR
C1 = 'beginnen met recordnr...'
C2 = 'eindigen met recordnr...'
C3 = 'hoogste recordnr in bestand...'
STORE 0 TO NR1,NR2,LNR1,LNR2
STOPPEN = .F.
IF VLAG
    DO INSTR WITH VLAG,LYNEN,KANTLIJN,MERK,CONDENS
ENDIF
USE &BRONF
DO WHILE .NOT. STOPPEN
    DO TWEEANT WITH NR1,NR2,C1,C2,C3
    LNR1 = LEN(LTRIM(STR(NR1)))
    LNR2 = LEN(LTRIM(STR(NR2)))
    BLZ = 0
    SET CONSOLE OFF
    SET PRINT ON
    DO BLAD4 WITH BLZ,LYNEN,NR1,NR2,PAPON,PAPUT,LNR1,LNR2
    SET PRINT OFF
    SET CONSOLE ON
    CLEAR
    JN = '*'
    DO WHILE UPPER(JN) <> 'J' .AND. UPPER(JN) <> 'N'
        @19,0
        WAIT SPACE(20)+'Nog een selectie?      <j/n>      ' TO JN
    ENDDO
    OPDR = IIF(UPPER(JN)='N','STOPPEN = .T.', '')
    &OPDR
ENDDO

```

```
CLOSE ALL
RETURN
```

Dit onderdeel van het programma zorgt voor het afdrukken van een konsekutieve reeks steekkaarten, die geselecteerd worden op grond van het volgnr van het eerste en laatste record uit de reeks.

```
***** PROG.PRG
```

```
*
```

```
RESTORE FROM WAARDEN.MEM
```

```
SET TALK OFF
```

```
SET BELL OFF
```

```
SET STATUS OFF
```

```
IF .NOT. FILE(STDf)
```

```
    CREATE &STDf FROM &OPBF
```

```
    INDEX ON AUTEUR TO &ANDX
```

```
    INDEX ON DRAGER TO &DNDX
```

```
    INDEX ON PLAATS TO &PNDX
```

```
    INDEX ON AARD TO &SNDX
```

```
    CLOSE ALL
```

```
ENDIF
```

```
ON ERROR DO FOUT1 WITH ERROR()
```

```
DO INFO WITH DELAY_1
```

```
DO WHILE .NOT. EINDE
```

```
    DO INITO WITH (STDf), (ANDX), (DNDX), (PNDX), (SNDX), TUS_F, WRKF, ;
```

```
        OPBF, NFILE, NDX_1, NDX_2, NDX_3, NDX_4, EINDE, DELAY_1
```

```
    OPDR = IIF(EINDE, 'EXIT', '')
```

```
    &OPDR
```

```
    DO MENU00 WITH TUS_F, NFILE, NDX_1, NDX_2, NDX_3, NDX_4, VLAG, PLN, ;
```

```
        KANTLIJN, MERK, CONDENS, EINDE, WRKF, SYSDRV, DETPAPON, ;
```

```
        DETPAPUT
```

```
ENDDO
```

```
SET STATUS ON
```

```
SET BELL ON
```

```
SET TALK ON
```

```
IF .NOT. VLAG
```

```
    SET CONSOLE OFF
```

```
    SET PRINT ON
```

```
    ?? NORMAAL
```

```
    SET MARGIN TO 0
```

```
    SET PRINT OFF
```

```
    SET CONSOLE ON
```

```
ENDIF
```

```
DO SLUTT
```

```
RETURN
```

Deze module is de ruggegraat van het programma. Hieraan zitten alle andere modules vast.

```

***** PRT.PRG
*
PARAMETER KEUZE,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,LYNEN,;
          PAPON,PAPUT,VLAG,KANTLIJN,MERK,CONDENS,FLOPPY,WRKF
CLEAR
STORE '' TO KRIT
C1 = KEUZE+'...' (STOPPEN: ESC) '
C2 = 'F10 = vorige invoer'
C5 = 'zoeken via ' + KEUZE
IF VLAG
  DO INSTR WITH VLAG,LYNEN,KANTLIJN,MERK,CONDENS
ENDIF
USE &BRONF
SET INDEX TO &BNDX_1,&BNDX_2,&BNDX_3,&BNDX_4
OPDR = IIF(RANG=0,'SET ORDER TO','SET ORDER TO RANG')
&OPDR
DO WHILE .T.
  CLEAR
  DUMMY = KRIT
  DO EENANT WITH KRIT,DUMMY,C1,C2,N
  OPDR = IIF(READKEY()=12,'EXIT','')
  &OPDR
  SEEK "&KRIT"
  IF FOUND()
    CLEAR
    @10,INT((79-LEN(C5))/2) SAY "&C5"
    BLZ = 0
    SET CONSOLE OFF
    SET PRINT ON
    DO BLAD WITH BLZ,KEUZE,KRIT,LYNEN,PAPON,PAPUT
    SET PRINT OFF
    SET CONSOLE ON
  ELSE
    DO OPDR601 WITH KRIT
  ENDIF
ENDDO
CLOSE ALL
RETURN

```

Deze module zorgt voor het afdrukken van de records die geselecteerd werden op grond van de inhoud van het veld AUTEUR, DRAGER, PLAATS of AARD.

```

***** PTR.PRG
*
PARAMETER KEUZE,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,LYNEN,;
          PAPON,PAPUT,VLAG,KANTLIJN,MERK,CONDENS,FLOPPY,WRKF
C1 = KEUZE+'...' (STOPPEN: ESC) '
C2 = 'F10 = vorige invoer'
STORE '' TO KRIT,DUMMY
IF VLAG
  DO INSTR WITH VLAG,LYNEN,KANTLIJN,MERK,CONDENS
ENDIF
USE &BRONF

```

```

DO WHILE .T.
  DO EENANT WITH KRIT,DUMMY,C1,C2,N
  DUMMY = KRIT
  OPDR = IIF(READKEY()=12,'EXIT','')
  &OPDR
  CLEAR
  @8,20 SAY 'Zoeken via '+KEUZE
  GO TOP
  LOCATE FOR "&KRIT"$TREFWRD
  IF FOUND()
    SET CONSOLE OFF
    SET PRINT ON
    BLZ = 0
    DO BLAD5 WITH BLZ,KEUZE,KRIT,LYNEN,PAPON,PAPUT
    SET PRINT OFF
    SET CONSOLE ON
  ELSE
    DO OPDR601 WITH KRIT
  ENDIF
ENDDO
CLOSE DATABASES
RETURN

```

Dit onderdeel stuurt het afdrukken van die steekkaarten
 waarvan het veld TREFWRD een door de gebruiker opgegeven
 trefwoord bevat.

```

***** REDSCHRM.PRG
*
PARAMETER MAUTEUR,MTITEL,MDRAGER,MPLAATS,MAARD,MTREFWRD
@ 0,0 CLEAR TO 22,79
SET SCOREBOARD OFF
@ 0,0 SAY 'Auteur      :'
@ 0,11 GET MAUTEUR FUNCTION "!S30"
@ 2,0 SAY 'Titel      :'
@ 2,11 GET MTITEL FUNCTION "!S60"
@ 4,0 SAY 'Drager     :'
@ 4,11 GET MDRAGER FUNCTION "!S60"
@ 6,0 SAY 'Plaats     :'
@ 6,11 GET MPLAATS FUNCTION "!S30"
@ 6,50 SAY 'Aard: '
@ 6,60 GET MAARD PICTURE "!"
@ 9,0 SAY 'Trefwoord  :'
@ 9,11 GET MTREFWRD FUNCTION "!S60"
@13,7 SAY "Cursor beweging      Wissen      Einde"
@15,11 SAY CHR(024)+"      HOME      letter:  Del      Opslaan: ^END"
@16,8 SAY CHR(027)+SPACE(5)+CHR(026)
@16,17 SAY "END      woord :      ^T      ^W"
@17,11 SAY CHR(025)+"      veld :      ^Y      Afbreken: ESC"
@12,5 TO 18,68 DOUBLE
@14,6 TO 14,67
READ
SET SCOREBOARD ON
RETURN

```

Deze module toont een steekkaart op het scherm. In dit scherm is het mogelijk de inhoud van de velden te veranderen.

***** SLUIT.PRG

*

CLEAR

@ 4,0

TEXT

Je verliet het programma

Je kunt

of terug via

DO PROG <CRT>

of terug naar dBASE via

nnnnn <CRT> met nnnnn een

dBASE opdracht

of terug naar DOS via

QUIT <CRT>

ENDTEXT

@ 3,20 TO 15,60

RETURN

Deze module verstrekt de gebruiker enige informatie bij het verlaten van het programma.

***** SPLIT.PRG

*

PARAMETER POS, POS1, TOETS, LENGTE

STORE 67 TO POS, POS1

STORE 0 TO BREEK

DO WHILE BREEK = 0

ON ERROR EXIT

POS = POS - 10

MTYD = SUBSTR("&TOETS", POS+1, 10)

BREEK = AT(" ", "&MTYD")

ENDDO

POS = POS+BREEK

IF (LENGTE - POS) > 67

BREEK = 0

DO WHILE BREEK = 0

POS1 = POS1 - 10

MTYD = SUBSTR("&TOETS", POS+POS1+1, 10)

BREEK = AT(" ", "&MTYD")

ENDDO

POS1 = POS1+BREEK

ELSE

POS1 = LENGTE - POS

ENDIF

RETURN

Zo de inhoud van de velden TITEL, DRAGER of TREFWRD te lang is om in het leesscherf op één lijn voorgesteld te worden, wordt met behulp van dit programmaonderdeel de inhoud netjes opgebroken ter hoogte van een spatie tussen twee woorden. Indien nodig wordt de inhoud van het veld in drie lijnen weergegeven.

***** TANK.PRG

*

```
PARAMETER FLOPPY,WISSEL,TUS_F
STORE " " TO NDX_1,NDX_2,NDX_3,NDX_4
DOELF = FLOPPY+"&TUS_F"
DO NDXEN WITH DOELF,NDX_1,NDX_2,NDX_3,NDX_4
IF .NOT. FILE(DOELF)
  CREATE &DOELF FROM OPBOUW
  INDEX ON AUTEUR TO &NDX_1
  INDEX ON DRAGER TO &NDX_2
  INDEX ON PLAATS TO &NDX_3
  INDEX ON AARD TO &NDX_4
  CLOSE ALL
ENDIF
DO CASE
  CASE WISSEL = 2
    DO MENU06 WITH DOELF,NDX_1,NDX_2,NDX_3,NDX_4
  CASE WISSEL = 4
    CLOSE ALL
    ERASE &DOELF
    ERASE &NDX_1
    ERASE &NDX_2
    ERASE &NDX_3
    ERASE &NDX_4
    LNF = LEN("&DOELF")
    DOELF=STUFF("&DOELF",LNF,1,"T")
    ERASE &DOELF
    RETURN TO MASTER
ENDCASE
RETURN
```

Het bestaan van een hulpbestand bij het invoeren van de gegevens wordt gecontroleerd. Is dit niet voorhanden, dan wordt het gekreëerd. Vervolgens worden de indexbestanden aangemaakt. Nadat de gegevens overgetapt zijn naar het permanente bestand kunnen via deze module de tijdelijke .dbf, .dbt en indexbestanden geschrapt worden.

***** TRANSFER.PRG

*

```
PARAMETER BRONF,DOELF,NDX_1,NDX_2,NDX_3,NDX_4, ;
      VLAG,PLN,KANTLIJN,MERK,CONDENS,FLOPPY
BRONF = FLOPPY+"&BRONF"
STORE " " TO MAUTEUR,MTITEL,MDRAGER,MPLAATS,MAARD,MIREFWRD
```

```

IF VLAG
  DO INSTR WITH VLAG, PLN, KANTLLJN, MERK, CONDENS
ENDIF
SELECT 1
USE &DOELF
SET INDEX TO &NDX_1, &NDX_2, &NDX_3, &NDX_4
SELECT 2
USE &BRONF
DO WHILE .NOT. EOF()
  KONTROLE = .T.
  MAUTEUR = AUTEUR
  MTITEL = TITEL
  MDRAGER = DRAGER
  MPLAATS = PLAATS
  MAARD = AARD
  MTREFWRD = TREFWRD
  DO CASE
    CASE AARD = "A" .OR. AARD = "F"
      RANG = 2
      TEST = MDRAGER
    CASE AARD = "B"
      RANG = 1
      TEST = MAUTEUR
    CASE AARD = "C"
      RANG = 1
      TEST = MAUTEUR
    CASE AARD = "D"
      RANG = 2
      TEST = MDRAGER
    OTHERWISE
      RANG = 2
      TEST = MDRAGER
  ENDCASE
  SELECT 1
  SET ORDER TO RANG
  SEEK "&TEST"
  IF FOUND()
    KONTROLE = .F.
    DO CASE
      CASE MAARD = "A" .OR. MAARD = "F"
        KRIT = AUTEUR
        KRITB = MAUTEUR
        PROEF = "DRAGER"
      CASE MAARD = "B"
        KRIT = TITEL
        KRITB = MTITEL
        PROEF = "AUTEUR"
      CASE MAARD = "C"
        KRIT = TITEL
        KRITB = MTITEL
        PROEF = "AUTEUR"
      CASE MAARD = "D"
        KRIT = AUTEUR
        KRITB = MAUTEUR
        PROEF = "DRAGER"
    ENDCASE
    DO WHILE &PROEF = "&TEST" .AND. .NOT. EOF()
      IF KRIT = KRITB

```

```

        EXIT
    ENDIF
    SKIP
    IF EOF()
        KONTOLE = .T.
        EXIT
    ENDIF
    ENDDO
ELSE
    KONTOLE = .T.
ENDIF
SELECT 2
IF .NOT. KONTOLE
    SET PRINT ON
    DO DRUKREC
    DELETE
    SET PRINT OFF
ENDIF
SKIP
ENDDO
SELECT 2
USE
SELECT 1
SET DELETED ON
APPEND FROM &BRONF
SET DELETED OFF
CLOSE ALL
RETURN

```

Dit programmaonderdeel schrijft de steekkaarten over van het tijdelijke naar het permanente bestand. Daarbij wordt een controle uitgevoerd op het eventueel reeds voorhanden zijn van de betrokken gegevens. Dit nazicht gebeurt aan de hand van een dubbel criterium. Blijkt konform de controlecriteria een record voor te komen, dan wordt de inhoud van deze steekkaart afgedrukt.

```

***** TWEEANT.PRG
*
PARAMETER KRIT1,KRIT2,C1,C2,C3
CLEAR
JN = '*'
OK = .F.
KRIT1 = 0
@ 5,10 SAY '&C3'+STR(RECCOUNT())
@ 7,10 SAY '&C1'
DO WHILE .NOT. OK
    @ 7,50 GET KRIT1 PICTURE "99999" RANGE 1,RECCOUNT()
    READ
    DO WHILE UPPER(JN) <> 'J' .AND. UPPER(JN) <> 'N'
        @9,0
        WAIT SPACE(10)+'Is dit gegeven OK?      <j/n>      ' TO JN
    ENDDO
    IF UPPER(JN) <> 'J'
        JN = '*'
    ENDIF

```

```

ELSE
    OK = .T.
ENDIF
ENDDO
@12,10 SAY '&C2'
JN = '*'
KRIT2 = 0
OK = .F.
DO WHILE .NOT. OK
    @12,50 GET KRIT2 PICTURE "99999" RANGE 1,RECCOUNT()
    READ
    DO WHILE UPPER(JN) <> 'J' .AND. UPPER(JN) <> 'N'
        @14,0
        WAIT SPACE(10)+'Is dit gegeven OK?      <j/n>      ' TO JN
    ENDDO
    IF UPPER(JN) <> 'J'
        JN = '*'
    ELSE
        OK = .T.
    ENDIF
ENDDO
CLEAR
RETURN

```

Deze module vraagt de gebruiker om twee gegevens in te voeren, bv het volgnr van het eerste en het laatste record van een reeks konsekutieve steekkaarten.

```

***** U_SCHRP.PRG
*
CLEAR
DO LSSCHRM
@ ROW()+2,0
DO WHILE .T.
    WAIT SPACE(10)+'Mag deze steekkaart geschrapt worden?';
    + '      <J/N>      'TO JN
    OPDR = IIF(AT('&JN','JN')=0,'LOOP','EXIT')
    &OPDR
ENDDO
OPDR = IIF(UPPER('&JN')='N','RECALL','')
&OPDR
RETURN

```

Deze module plaatst een boodschap voor de gebruiker op het scherm en vereist een antwoord.

```

***** VERVANG.PRG
*
PARAMETER GEW,REC,RANG,X1,X2,X3,X4,X5,X6
GEW = .T.
SELECT 1

```

```

OPDR = IIF(RANG = 0, 'SET ORDER TO 1', '')
&OPDR
GOTO REC
REPLACE AUTEUR WITH X1, TITEL WITH X2, DRAGER WITH X3, ;
PLAATS WITH X4, AARD WITH X5, TREFWRD WITH X6
OPDR = IIF(RANG = 0, 'SET ORDER TO', '')
&OPDR
RETURN

```

Deze module regelt het vervangen van een record in het bestand nadat de gebruiker de inhoud van één of ander veld, het memoveld uitgezonderd, wijzigde.

***** WIS.PRG

*

```

PARAMETER KEUZE, RANG, N, BRONF, BNDX_1, BNDX_2, BNDX_3, BNDX_4, LYNEN, ;
          PAPON, PAPUT, VLAG, KANTLIJN, MERK, CONDENS, FLOPPY, WRKF

```

```

PRIVATE Z, KOL

```

```

CLEAR

```

```

C1 = KEUZE+"...      (STOPPEN: ESC)"

```

```

C2 = "F10 = vorige invoer"

```

```

STORE '' TO KRIT, DUMMY, KOM1

```

```

USE &BRONF

```

```

SET INDEX TO &BNDX_1, &BNDX_2, &BNDX_3, &BNDX_4

```

```

OPDR = IIF(RANG=0, "SET ORDER TO", "SET ORDER TO RANG")

```

```

&OPDR

```

```

SELECT 2

```

```

USE &WRKF

```

```

DO WHILE .T.

```

```

    DO EENANT WITH KRIT, DUMMY, C1, C2, N

```

```

    OPDR = IIF(READKEY()=12, 'EXIT', '')

```

```

    &OPDR

```

```

    DUMMY = KRIT

```

```

    KOM = KEUZE+'...      &KRIT'

```

```

    Z = LEN(KOM)

```

```

    KOL = INT((79-Z)/2)

```

```

    @23, KOL SAY '&KOM'

```

```

    SELECT 1

```

```

    SEEK '&KRIT'

```

```

    IF FOUND()

```

```

        Z3 = LEN(KRIT)

```

```

        DO WHILE KRIT = LEFT(&KEUZE, Z3)

```

```

            REC = RECNO()

```

```

            SELECT 2

```

```

            APPEND BLANK

```

```

            REPLACE RECNR WITH REC

```

```

            SELECT 1

```

```

            SKIP

```

```

        ENDDO

```

```

    SELECT 2

```

```

    GO TOP

```

```

    REC = RECNR

```

```

    KOL1 = 0

```

```

    LEERSTE = .T.

```

```

    LLAATSTE = IIF(RECNO() = RECCOUNT(), .T., .F.)

```

```

ELSE
    OK = .T.
ENDIF
ENDDO
@12,10 SAY '&C2'
JN = '*'
KRIT2 = 0
OK = .F.
DO WHILE .NOT. OK
    @12,50 GET KRIT2 PICTURE "99999" RANGE 1,RECCOUNT()
    READ
    DO WHILE UPPER(JN) <> 'J' .AND. UPPER(JN) <> 'N'
        @14,0
        WAIT SPACE(10)+'Is dit gegeven OK?      <j/n>      ' TO JN
    ENDDO
    IF UPPER(JN) <> 'J'
        JN = '*'
    ELSE
        OK = .T.
    ENDIF
ENDDO
CLEAR
RETURN

```

Deze module vraagt de gebruiker om twee gegevens in te voeren, bv het volgnr van het eerste en het laatste record van een reeks konsekutieve steekkaarten.

```

***** U_SCHRP.PRG
*
CLEAR
DO LSSCHRM
@ ROW()+2,0
DO WHILE .T.
    WAIT SPACE(10)+'Mag deze steekkaart geschrapt worden?';
    + '      <J/N>      'TO JN
    OPDR = IIF(AT('&JN','JN')=0,'LOOP','EXIT')
    &OPDR
ENDDO
OPDR = IIF(UPPER('&JN')='N','RECALL','')
&OPDR
RETURN

```

Deze module plaatst een boodschap voor de gebruiker op het scherm en vereist een antwoord.

```

***** VERVANG.PRG
*
PARAMETER GEW,REC,RANG,X1,X2,X3,X4,X5,X6
GEW = .T.
SELECT 1

```

```

OPDR = IIF(RANG = 0, 'SET ORDER TO 1', '')
&OPDR
GOTO REC
REPLACE AUTEUR WITH X1, TITEL WITH X2, DRAGER WITH X3, ;
PLAATS WITH X4, AARD WITH X5, TREFWRD WITH X6
OPDR = IIF(RANG = 0, 'SET ORDER TO', '')
&OPDR
RETURN

```

Deze module regelt het vervangen van een record in het bestand nadat de gebruiker de inhoud van één of ander veld, het memoveld uitgezonderd, wijzigde.

```

***** WIS.PRG
*
PARAMETER KEUZE, RANG, N, BRONF, BNDX_1, BNDX_2, BNDX_3, BNDX_4, LYNEN, ;
          PAPON, PAPUT, VLAG, KANTLLJN, MERK, CONDENS, FLOPPY, WRKF
PRIVATE Z, KOL
CLEAR
C1 = KEUZE+"...      (STOPPEN: ESC)"
C2 = "F10 = vorige invoer"
STORE '' TO KRIT, DUMMY, KOM1
USE &BRONF
SET INDEX TO &BNDX_1, &BNDX_2, &BNDX_3, &BNDX_4
OPDR = IIF(RANG=0, "SET ORDER TO", "SET ORDER TO RANG")
&OPDR
SELECT 2
USE &WRKF
DO WHILE .T.
  DO EENANT WITH KRIT, DUMMY, C1, C2, N
  OPDR = IIF(READKEY()=12, 'EXIT', '')
  &OPDR
  DUMMY = KRIT
  KOM = KEUZE+'...      &KRIT'
  Z = LEN(KOM)
  KOL = INT((79-Z)/2)
  @23, KOL SAY '&KOM'
  SELECT 1
  SEEK '&KRIT'
  IF FOUND()
    Z3 = LEN(KRIT)
    DO WHILE KRIT = LEFT(&KEUZE, Z3)
      REC = RECNO()
      SELECT 2
      APPEND BLANK
      REPLACE RECNR WITH REC
      SELECT 1
      SKIP
    ENDDO
  SELECT 2
  GO TOP
  REC = RECNR
  KOL1 = 0
  LEERSTE = .T.
  LLAATSTE = IIF(RECNO() = RECCOUNT(), .T., .F.)

```

```

STOPPEN = .F.
DO WHILE .NOT. STOPPEN
  SELECT 1
  GOTO REC
  DO KOMTR 1 WITH KOM1,KOL1,LEERSTE,LLAATSTE
  C3 = IIF(DELETED(),'logisch geschrapt','')
  Z3 = LEN(C3)
  KOL3 = INT((79-Z3)/2)
  @2,0 CLEAR TO 3,79
  @2,KOL1 SAY "&KOM1"
  @3,KOL3 SAY "&C3"
  DO LSSCHRM
  ANT = '*'
  @21,10 SAY 'A(nte P(ost W(is O(pvissen E(xit'
  DO WHILE .T.
    @21,70 GET ANT PICTURE "!"
    READ
    OPDR = IIF(AT('&ANT','APWOE')=0,'LOOP','EXIT')
    &OPDR
  ENDDO
  DO WISSEL_4 WITH ANT,REC,STOPPEN,LEERSTE,LLAATSTE
ENDDO
SELECT 2
DELETE ALL
PACK
SELECT 1
ELSE
  DO OPDR601 WITH KRIT
ENDIF
ENDDO
CLEAR
CLOSE ALL
RETURN

```

Dit onderdeel laat toe om geselecteerde records logisch te schrappen of om zo'n schrappen ongedaan te maken. De selectie gebeurt op grond van de AUTEUR, de DRAGER, de PLAATS of de AARD.

***** WISAND.PRG

*

PARAMETER KEUZE,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,LYNEN, ;
 PAPON,PAPUT,VLAG,KANTILIJN,MERK,CONDENS,FLOPPY,WRF

CLEAR

Z5 = LEN(KEUZE)

KOL5 = INT((79-Z5)/2)

STORE '' TO KRIT,KOM1

USE &BRONF

SELECT 2

USE &WRF

SELECT 1

DO WHILE .T.

CLEAR

DO WHILE .T.

@8,0

```

WAIT SPACE(20) + '... STOPPEN? <J/N> ' TO X
OPDR = IIF("&X"$"JjNn", 'EXIT', '')
&OPDR
ENDDO
OPDR = IIF("&X"$"Jj", "EXIT", "")
&OPDR
DO OPDR604
@22, KOL5 SAY '&KEUZE'
SET FILTER TO FILE LIT
LOCATE
IF FOUND()
DO WHILE .NOT. EOF()
REC = RECNO()
SELECT 2
APPEND BLANK
REPLACE RECNR WITH REC
SELECT 1
SKIP
ENDDO
SELECT 2
GO TOP
REC = RECNR
KOL1 = 0
LEERSTE = .T.
LLAATSTE = IIF(RECNO()=RECCOUNT(), .T., .F.)
STOPPEN = .F.
CLEAR
DO WHILE .NOT. STOPPEN
@22, KOL5 SAY "&KEUZE"
SELECT 1
GOTO REC
DO KOM1_1 WITH KOM1, KOL1, LEERSTE, LLAATSTE
C3 = IIF(DELETED(), 'logisch geschrappt', '')
Z3 = LEN('&C3')
KOL3 = INT((79-Z3)/2)
@2, 0 CLEAR TO 3, 79
@2, KOL1 SAY "&KOM1"
@3, KOL3 SAY "&C3"
DO LSSCHRM
@21, 10 SAY 'A(nte P(ost W(is O(pvissen E(xit'
ANT = '*'
DO WHILE .T.
@21, 70 GET ANT PICTURE '!'
READ
OPDR = IIF(AT('&ANT', 'APWOE')=0, 'LOOP', 'EXIT')
&OPDR
ENDDO
DO WISSEL_4 WITH ANT, REC, STOPPEN, LEERSTE, LLAATSTE
ENDDO
SELECT 2
DELETE ALL
PACK
SELECT 1
SET FILTER TO
ELSE
CLEAR
DO OPDR601 WITH KRIT
ENDIF

```

```

ENDDO
CLOSE ALL
RETURN

```

Dit onderdeel laat toe om geselecteerde records logisch te schrappen of om zo'n schrappen ongedaan te maken. De selectie gebeurt op grond van de QUERY mogelijkheden van dBASEIII+.

```

***** WISNR.PRG

```

```

*

```

```

PARAMETERS KEUZE,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,LYNEN,;
          PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,WRKF

```

```

CLEAR

```

```

C1 = 'recordnr...      (STOPPEN: ESC)'

```

```

C2 = 'F10 = vorige invoer'

```

```

KOL0 = INT((79-LEN(KEUZE))/2)

```

```

STORE '' TO KRIT,DUMMY

```

```

DO WHILE .T.

```

```

    DO EENANT WITH KRIT,DUMMY,C1,C2,N

```

```

    OPDR = IIF(READKEY()=12,'EXIT','')

```

```

    &OPDR

```

```

    DUMMY = KRIT

```

```

    USE &BRONF

```

```

    IF VAL("&KRIT") > RECCOUNT()

```

```

        CLEAR

```

```

        @10,10 SAY 'Het hoogste recordnr is '+STR(RECCOUNT())

```

```

        @12,0

```

```

        SET ESCAPE OFF

```

```

        WAIT SPACE(20)+'Druk willekeurige toets...      ' TO ANT

```

```

        SET ESCAPE ON

```

```

        LOOP

```

```

    ENDIF

```

```

    GOTO RECORD VAL("&KRIT")

```

```

DO WHILE .T.

```

```

    C4 = IIF(DELETED(),'logisch geschrapt','')

```

```

    Z4 = LEN(C4)

```

```

    KOL4 = INT((79-Z4)/2)

```

```

    @2,0 CLEAR TO 3,79

```

```

    @2,KOL4 SAY "&C4"

```

```

    @22,KOL0 SAY "&KEUZE"

```

```

    DO LSSCHRM

```

```

    @21,15 SAY 'W(is O(pvissen E(xit'

```

```

    ANT = '*'

```

```

DO WHILE .T.

```

```

    @21,55 GET ANT PICTURE '!'

```

```

    READ

```

```

    OPDR = IIF(AT('&ANT','WOE')=0,'LOOP','EXIT')

```

```

    &OPDR

```

```

ENDDO

```

```

DO CASE

```

```

    CASE ANT = 'E'

```

```

        EXIT

```

```

    CASE ANT = 'O'

```

```

        RECALL

```

```

CASE ANT = 'W'
DELETE
ENDCASE
ENDDO
CLOSE ALL
ENDDO
RETURN

```

Dit onderdeel laat toe om een geselecteerd record logisch te schrappen of om zo'n schrappen ongedaan te maken. De selectie gebeurt op grond van het volgnummer van het record in het fysieke bestand.

***** WISNRS.PRG

*

```

PARAMETERS KEUZE,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,LYNEN,;
PAPON,PAPUT,VLAG,KANTLLIJN,MERK,CONDENS,FLOPPY,WRKF

```

CLEAR

C1 = 'beginnen met recordnr...'

C2 = 'eindigen met recordnr...'

C3 = 'laatste recordnr in bestand...'

STOPPEN = .F.

STORE 0 TO NR1,NR2,KOL2

KOM2 = ''

Z5 = LEN(KEUZE)

KOL5 = INT((79-Z5)/2)

DO WHILE .NOT. STOPPEN

USE &BRONF

DO TWEEANT WITH NR1,NR2,C1,C2,C3

GOTO RECORD NR1

LEERSTE = .T.

LLAATSTE = IIF(RECNO()=NR2,.T.,.F.)

@22,KOL5 SAY '&KEUZE'

DO WHILE RECNO() <= NR2 .AND. RECNO() >= NR1

C4 = IIF(DELETED(),'logisch geschrapt','')

Z4 = LEN(C4)

KOL4 = INT((79-Z4)/2)

DO KOMTR_1 WITH KOM2,KOL2,LEERSTE,LLAATSTE

@2,0 CLEAR TO 3,79

@2,KOL2 SAY "&KOM2"

@3,KOL4 SAY "&C4"

DO LSSCHRM

@21,10 SAY 'A(nte P(ost W(is O(pvissen E(xit'

ANT = '*'

DO WHILE .T.

@21,70 GET ANT PICTURE '!!'

READ

OPDR = IIF(AT('&ANT','APWOE')=0,'LOOP','EXIT')

&OPDR

ENDDO

DO CASE

CASE ANT = 'E'

EXIT

CASE ANT = 'O'

RECALL

```

CASE ANT = 'W'
  DELETE
CASE ANT = 'P'
  IF RECNO() < NR2
    SKIP 1
  ENDIF
CASE ANT = 'A'
  IF RECNO() > NR1
    SKIP-1
  ENDIF
ENDCASE
LEERSTE = IIF(RECNO()=NR1,.T.,.F.)
LLAATSTE = IIF(RECNO()=NR2,.T.,.F.)
ENDDO
CLEAR
JN = '*'
DO WHILE UPPER(JN) <> 'J' .AND. UPPER(JN) <> 'N'
  @19,0
  WAIT SPACE(20)+'Nog een selectie?      <j/n>      ' TO JN
ENDDO
IF UPPER(JN) = 'N'
  STOPPEN = .T.
ENDIF
ENDDO
CLOSE ALL
RETURN

```

Dit onderdeel laat toe om geselecteerde records logisch te schrappen of om zo'n schrappen ongedaan te maken. De selectie gebeurt op grond van het volgnummer van het eerste en laatste te behandelen record.

***** WISSEL_1.PRG

*

PARAMETER X,GEW,REC,STOP,MAUTEUR,MTITEL,MDRAGER,MPLAATS, ;
 MAARD,MTREFWRD,RANG

DO CASE

CASE X = 'A'

GEW = .F.

SELECT 2

SKIP -1

OPDR = IIF(BOF(),'GO TOP','')

&OPDR

REC = RECNR

CASE X = 'P'

GEW = .F.

SELECT 2

SKIP

OPDR = IIF(EOF(),'SKIP -1','')

&OPDR

REC = RECNR

CASE X = 'W'

MAUTEUR = AUTEUR

MTITEL = TITEL

MDRAGER = DRAGER

```

MPLAATS = PLAATS
MAARD = AARD
MTREFWRD = TREFWRD
DO REDSCHRM;
WITH MAUTEUR,MTITEL,MDRAGER,MPLAATS,MAARD,MTREFWRD
IF READKEY() = 270
    DO VERVANG WITH GEW,REC,RANG,MAUTEUR,MTITEL,MDRAGER,;
    MPLAATS,MAARD,MTREFWRD
ENDIF
SELECT 2
CASE X = 'S'
    GEW = .F.
    SELECT 1
    DELETE
    SELECT 2
CASE X = 'O'
    GEW = .F.
    SELECT 1
    RECALL
    SELECT 2
CASE X = 'E'
    STOP = .T.
CASE X = 'M'
    CHANGE FIELDS MEMO
    CLEAR
ENDCASE
RETURN

```

Dit onderdeel fungeert als een soort spoorwegwissel. Het verloop van het programma wordt in één van meerdere mogelijkheden gestuurd.

```

***** WISSEL_2.PRG
*
PARAMETER X,GEW,REC,RANG,MAUTEUR,MTITEL,MDRAGER,MPLAATS, ;
    MAARD,MTREFWRD,KRIT1,KRIT2
DO CASE
CASE X = 'A'
    GEW = .F.
    SKIP -1
    OPDR = IIF(RECNO()<KRIT1,'SKIP','')
    &OPDR
    REC = RECNO()
CASE X = 'P'
    GEW = .F.
    SKIP
    OPDR = IIF(RECNO()>KRIT2,'SKIP -1','')
    &OPDR
    REC = RECNO()
CASE X = 'W'
    MAUTEUR = AUTEUR
    MTITEL = TITEL
    MDRAGER = DRAGER
    MPLAATS = PLAATS
    MAARD = AARD

```

```

MIREFWRD = TREFWRD
DO REDSCHRM;
WITH MAUTEUR,MTITEL,MDRAGER,MPLAATS,MAARD,MIREFWRD
IF READKEY() = 270
    DO VERVANG WITH GEW,REC,RANG,MAUTEUR,MTITEL,MDRAGER,;
    MPLAATS,MAARD,MIREFWRD
ENDIF
CASE X = 'S'
    GEW = .F.
    DELETE
CASE X = 'O'
    GEW = .F.
    RECALL
CASE X = 'M'
    CHANGE FIELDS MEMO
    CLEAR
CASE X = 'E'
    REC = RECCOUNT() + 1
ENDCASE
RETURN

```

Deze module laat de selectie uit een meervoudige keuze toe.

```

***** WISSEL_3.PRG
*
PARAMETER X,GEW,REC,MAUTEUR,MTITEL,MDRAGER,MPLAATS,MAARD,;
    MIREFWRD,RANG
DO CASE
CASE X = 'S'
    GEW = .F.
    DELETE
CASE X = 'O'
    GEW = .F.
    RECALL
CASE X = 'W'
    MAUTEUR = AUTEUR
    MTITEL = TITEL
    MDRAGER = DRAGER
    MPLAATS = PLAATS
    MAARD = AARD
    MIREFWRD = TREFWRD
    DO REDSCHRM;
    WITH MAUTEUR,MTITEL,MDRAGER,MPLAATS,MAARD,MIREFWRD
    IF READKEY()=270
        DO VERVANG WITH GEW,REC,RANG,MAUTEUR,MTITEL,MDRAGER,;
        MPLAATS,MAARD,MIREFWRD
    ENDIF
CASE X = 'M'
    CHANGE FIELDS MEMO
CASE X = 'E'
ENDCASE
RETURN

```

Dit onderdeel laat een selectie uit een meervoudige keuze toe.

***** WISSEL_4.PRG

*

PARAMETERS ANT,REC,STOP,LEERSTE,LLAATSTE

DO CASE

CASE ANT = 'E'

STOP = .T.

CASE ANT = 'O'

RECALL

CASE ANT = 'W'

DELETE

CASE ANT = 'P'

SELECT 2

SKIP

IF EOF()

SKIP -1

ENDIF

REC = RECNR

LEERSTE = IIF(RECNO()=1,.T.,.F.)

LLAATSTE = IIF(RECNO()=RECCOUNT(),.T.,.F.)

CASE ANT = 'A'

SELECT 2

SKIP -1

REC = RECNR

LEERSTE = IIF(RECNO()=1,.T.,.F.)

LLAATSTE = IIF(RECNO()=RECCOUNT(),.T.,.F.)

ENDCASE

RETURN

Dit programmaonderdeel laat de selectie toe van één uit meerdere mogelijkheden.

***** WISTRF.PRG

*

PARAMETER KEUZE,RANG,N,BRONF,BNDX_1,BNDX_2,BNDX_3,BNDX_4,LYNEN,;
PAPON,PAPUT,VLAG,KANTLLJN,MERK,CONDENS,FLOPPY,WRKF

CLEAR

C1 = KEUZE+'..' (STOPPEN: ESC)'

C2 = 'F10 = vorige invoer'

STORE '' TO KRIT,DUMMY,KOM1

USE &BRONF

SELECT 2

USE &WRKF

DO WHILE .T.

SELECT 1

DO EENANT WITH KRIT,DUMMY,C1,C2,N

IF READKEY() = 12

EXIT

```

ENDIF
DUMMY = KRIT
KOM = KEUZE+'...' &KRIT'
Z0 = LEN(KOM)
KOL0 = INT((79-Z0)/2)
GO TOP
LOCATE FOR "&KRIT"$TREFWRD
IF FOUND()
    DO WHILE .NOT. EOF()
        REC = RECNO()
        SELECT 2
        APPEND BLANK
        REPLACE RECNR WITH REC
        SELECT 1
        CONTINUE
    ENDDO
    SELECT 2
    GO TOP
    REC = RECNR
    KOL1 = 0
    LEERSTE = .T.
    LLAATSTE = IIF(RECNO()=RECCOUNT(),.T.,.F.)
    STOPPEN = .F.
    DO WHILE .NOT. STOPPEN
        SELECT 1
        GOTO REC
        DO KOM1_1 WITH KOM1,KOL1,LEERSTE,LLAATSTE
        C3 = IIF(DELETED(),'logisch geschrapt','')
        Z3 = LEN(C3)
        KOL3 = INT((79-Z3)/2)
        @2,0 CLEAR TO 3,79
        @2,KOL1 SAY "&KOM1"
        @3,KOL3 SAY "&C3"
        DO LSSCHRM
        ANT = '*'
        @21,10 SAY 'A(nte P(ost W(is O(pvissen E(xit'
        DO WHILE .T.
            @21,70 GET ANT PICTURE '!'
            READ
            OPDR = IIF(AT('&ANT','APWOE')=0,'LOOP','EXIT')
            &OPDR
        ENDDO
        DO WISSEL_4 WITH ANT,REC,STOPPEN,LEERSTE,LLAATSTE
    ENDDO
    SELECT 2
    DELETE ALL
    PACK
    SELECT 1
ELSE
    DO OPDR601 WITH KRIT
ENDIF
ENDDO
CLOSE ALL
RETURN

```

Dit onderdeel laat toe om geselecteerde records logisch te schrappen of om zo'n schrappen ongedaan te maken. De

selectie gebeurt op grond van een door de gebruiker opgegeven trefwoord.
