

Flex_Rooster

WERKBOEK

RPG Deel 1

Dit werkboek is eigendom van ICS opleidingen en mag niet worden meegenomen.

©ICS Opleidingen

Niets uit deze uitgave mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm, geluidsband, elektronisch of op welke andere wijze dan ook zonder voorafgaande schriftelijke toestemming van ICS Opleidingen.

INHOUDSOPGAVE

1	SCHERMBESTANDEN IN RPG PROGRAMMA'S.....	3
1.1	Functietoetsen in een schermprogramma.....	6

1 SCHERMBESTANDEN IN RPG PROGRAMMA'S

Lees

Lees in het cursusboek het volgende hoofdstuk :
 Hoofdstuk 4 "FILE SPECIFICATIONS"

Lees ook nog een keer in dit werkboek paragraaf 3.1 File Specificaties Keywords

Een schermbestand vormt een onderdeel in de dialoog tussen programma/computer en de gebruiker.

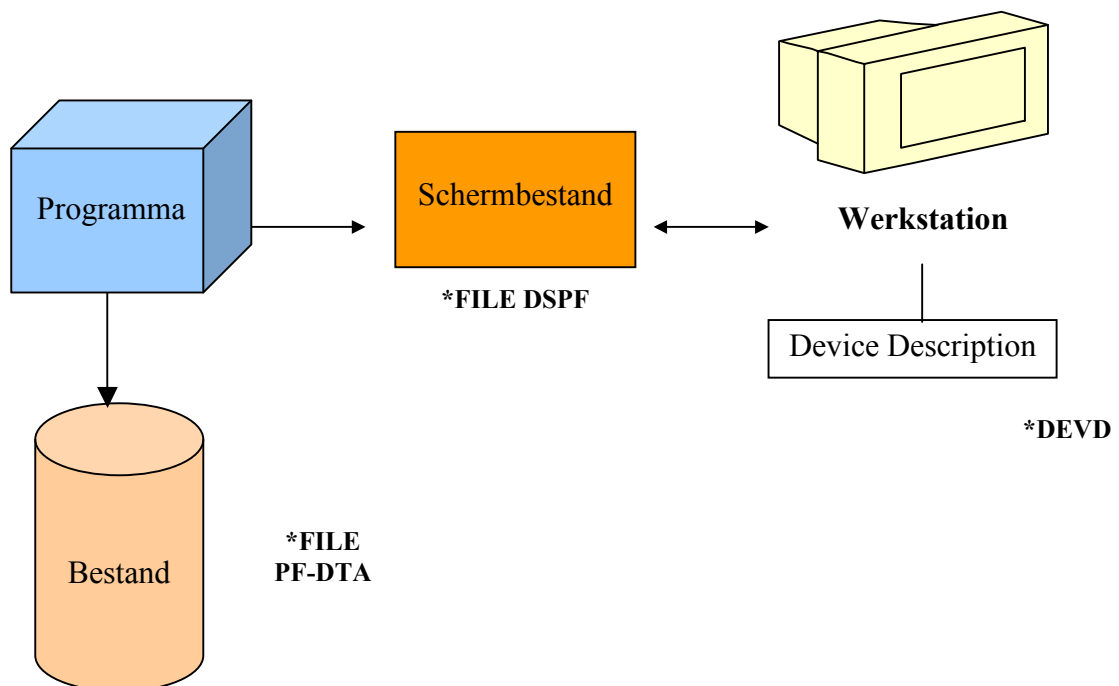
Een schermbestand beschrijft via een recordformaat de lay-out van het scherm, dat aan de gebruiker wordt getoond.

Ook bij schermbestanden wordt in het RPG-programma altijd gewerkt met een recordformaat. In een schermbestand kunnen overigens verschillende recordformaten voorkomen met elk een eigen lay-out of functie.

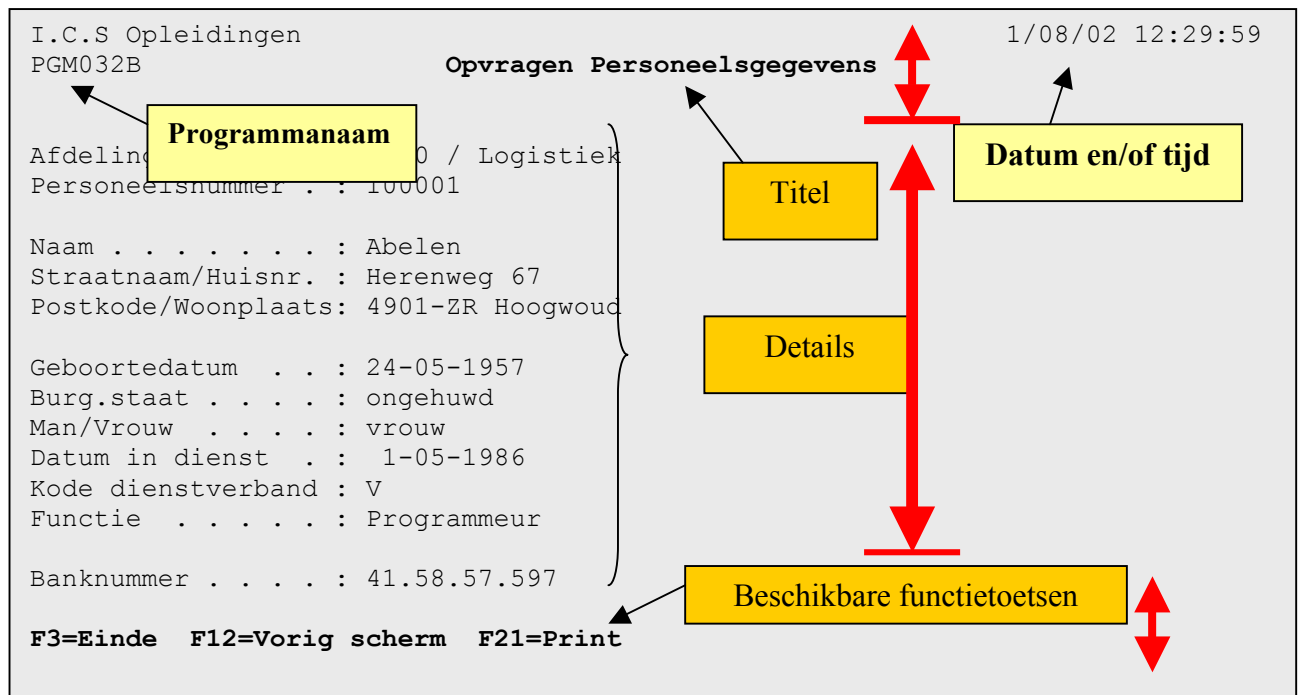
Bij een schermbestand wordt doorgaans uitgegaan van de grootte van een traditioneel werkstation. Dat is 24 regels of lijnen hoog bij 80 tekens breed. Soms wordt hier van afgeweken: 27 regels bij 132 tekens. Deze instellingen zijn afhankelijk het type werkstation.

Men kan ook het scherm kleiner maken dan de standaardgrootte. In dit geval spreekt men van een **window**.

In de oefeningen tijdens deze cursus gaan we uit van de normale grootte (24 bij 80) van het beeldscherm.



In de opbouw van een recordformaat voor een schermbestand zien we de volgende onderdelen:



- Het **bovengedeelte** met titel, datum, tijd, bedrijfsnaam, programmanaam etc.. is net als bij een lijst een **kopgedeelte**, dat in het recordformaat is gecodeerd.
De titel geeft weer welke functie wordt uitgevoerd met behulp van dit scherm. In dit geval worden de gegevens getoond van een personeelslid en wordt dit in de titel tot uitdrukking gebracht.
De juiste titel is belangrijk, omdat bij eventuele fouten of onduidelijkheden de helpdesk geraadpleegd kan worden. De gebruiker kan dan aangeven op welk scherm zijn vraag betrekking heeft. Daarbij wordt ook vaak de programmanaam gebruikt die dan ook in het bovengedeelte is opgenomen.
- Het **middengedeelte** is het gedeelte waar de gebruiker gegevens kan opvragen of invoeren.
- Het **ondergedeelte** bevat op een AS/400 altijd informatie over de functietoetsen die in het programma kunnen worden gebruikt.

Om een recordformaat te kunnen tonen moeten we het recordformaat vaak eerst initialiseren, zodat de te tonen velden gevuld zijn met gegevens.

Er wordt vaak één schermformaat gebruikt om de gebruiker gegevens te laten invoeren en een ander schermformaat om de gevraagde gegevens te tonen.

Bijvoorbeeld :

- Laat de gebruiker een selectie criterium of zoekgegeven intoetsen
- Laat het programma de gegevens aan de hand van de selectiecriteria/zoekgegevens verzamelen.
- Toon het resultaat aan de gebruiker.
- Herhaal deze handelingen tot de gebruiker er genoeg van heeft.

De instructie om een schermformaat aan de gebruiker te tonen is **EXFMT Execute Format**. Deze instructie is een combinatie van een *schrijfopdracht*, waarmee het schermbestand getoond wordt en een *leesopdracht*, waarmee het schermbestand wordt gelezen door het programma. Door deze combinatie, die altijd moet worden toegepast om een dialoog te kunnen hebben tussen computer en gebruiker, moet men een schermbestand als combined-file in de File-specificaties definiëren.

```

FFilename++IPEASF.....L.....A.Device+.Keywords+++++++
*
FRPGDET      CF      E              WORKSTN
  
```

Uiteraard wordt in het programma gebruik gemaakt van *iteratie* (=herhaling), zodat het invul scherm voor de selectiecriteria/zoekgegevens zolang wordt getoond tot de gebruiker het programma beëindigt.

In tegenstelling tot het lezen van een fysiek bestand met een **Do-While**, gebruiken we bij het werken met schermformaten bij voorkeur een **Do-Until**.

```

*
* stuur scherm totdat F3=Einde
*
C          *INKC   DOUEQ*ON
C          *INKC   EXFMTRPGDETR1
C          *INKC   IFEQ *OFF
*
* uitvoeren van de actie
*
C          EXSR ACTIE
*
C          ENDIF
C          ENDDO
*
* afsluiten programma
*
C          SETON           LR
*
  
```

Het uitsturen van een scherm in een **Do_Until** heeft het voordeel, dat het scherm altijd op slechts één plaats in het programma wordt getoond aan de gebruiker. Anders wordt het overzicht bij invulschermen met controles op de ingevoerde gegevens wellicht onmogelijk.

Dit zou men als een ongeschreven regel kunnen beschouwen.

Het correct specificeren van de functietoetsen (met name die waarmee de loop in het programma kan worden beëindigd) is van groot belang. Op de AS/400 is het vanaf het vroegste begin standaard om een programma te beëindigen met functietoets 3 <F3> .

Ook het tonen van de beschikbare functietoetsen onderin het scherm is ook een standaard, waar de AS/400 en de programmeurs zeer strikt in zijn.

1.1 Functietoetsen in een schermprogramma

Om de functietoetsen (<F1> t/m F24) te **mogen** gebruiken in een RPG-programma moeten ze in de DDS source van het schermbestand worden gespecificeerd .

Bijvoorbeeld voor het gebruik van functietoets <F3=Einde> wordt command-action **CA03** of command-function **CF03** opgenomen in de DDS-source.

Deze codering in het schermbestand correspondeert in het RPG-programma indicator **KC**, die ook als indicatorveld ***INKC** kan worden gebruikt.

Indicator in RPG	Functietoets DDS	Indicator in RPG	Functietoets DDS
KA (*INKA)	CA01/CF01 (F1)	KM (*INKM)	CA13/CF13 (F13)
KB (*INKB)	CA02/CF02 (F2)	KN (*INKN)	CA14/CF14 (F14)
KC (*INKC)	CA03/CF03 (F3)	KP (*INKP)	CA15/CF15 (F15)
KD (*INKD)	CA04/CF04 (F4)	KQ (*INKQ)	CA16/CF16 (F16)
KE (*INKE)	CA05/CF05 (F5)	KR (*INKR)	CA17/CF17 (F17)
KF (*INKF)	CA06/CF06 (F6)	KS (*INKS)	CA18/CF18 (F18)
KG (*INKG)	CA07/CF07 (F7)	KT (*INKT)	CA19/CF19 (F19)
KH (*INKH)	CA08/CF08 (F8)	KU (*INKU)	CA20/CF20 (F20)
KI (*INKI)	CA09/CF09 (F9)	KV (*INKV)	CA21/CF21 (F21)
KJ (*INKJ)	CA10/CF10 (F10)	KW (*INKW)	CA22/CF22 (F22)
KK (*INKK)	CA11/CF11 (F11)	KX (*INKX)	CA23/CF23 (F23)
KL (*INKL)	CA12/CF12 (F12)	KY (*INKY)	CA24/CF24 (F24)

Tabel 12-2 Functietoets Indicatoren en Corresponderende Functietoetsen.

Merk op dat KO en KZ niet wordt gebruikt !

Bij het specificeren van de functietoetsen in het schermbestand worden de codes CAxx/CFxx vaak voorzien van een van de werk- of vrije indicatoren 01 t/m 99.

Dat houdt in dat bij gebruik van bijvoorbeeld functietoets <F1>, niet alleen indicator KA wordt “AAN” gezet maar ook de opgegeven vrije indicator.

Standaards voor het gebruik van vrije indicatoren hangt af van de standaardregels, die men per bedrijf heeft vastgelegd en die kunnen van bedrijf tot bedrijf verschillen.

Wel ziet men vaak dat de indicatoren 01 t/m 24 zijn gereserveerd voor het gebruik bij functietoetsen. Treft men in zo'n programma het gebruik van indicator 19 aan, dan weet men dat deze overeenkomt met het gebruik van functietoets <F19>.

Voor het uitsturen van elk willekeurig schermformaat kunnen we gebruik maken van de bovenstaande structuur. Omdat we in ons voorbeeld te maken hebben met twee schermformaten maken we gebruik van een subroutine voor het uitsturen van dat tweede schermformaat. Daarmee houden we het programma overzichtelijk.

Die subroutine kan er als volgt uitzien:

```

* stuur scherm-2 totdat F12=Terug
*
C      ACTIE      BEGSR
*
C      *INKL      DOUEQ*ON
C                      EXFMTRPGDETR2
C      *INKL      IFEQ *OFF
*
*
C                      ENDIF
C                      ENDDO
*
C                      ENDSR
  
```

Doe

In deze oefening maken we een basis programma, waarin de zojuist beschreven structuur is terug te vinden. Volg onderstaande stappen.

1. Maak een programma voor het opvragen van personeelsgegevens.
2. Noem dit programma **RPG180**.
3. Maak gebruik van het volgende schermbestand:
 - **RPGDET** Schermbestand
4. Zorg ervoor dat dit schermbestand aan de hand van de source wordt gecompileerd in uw eigen bibliotheek.
5. Neem voor het programma de zojuist beschreven voorbeelden van de RPG- source in dit hoofdstuk over.
6. Compileer het programma en voer het programma uit.