



Dansk Brugerhåndbog

Copyright © (1990-2022) SW-Tools ApS

Duevej 23

DK-3250 Solrød Strand

Denmark

Phone: +45) 33 33 05 56

Mail: swtools@swtools.com

www: www.swtools.com

Beregninger og subfunktioner

22/11/01 / 2022-09-01 008.384

Indholdsfortegnelse

Indholdsfortegnelse	2
1. Introduktion.....	6
1.1. Eksempler.....	7
1.1.1. <u>IF..ELSE</u> - Betingede sætninger.....	8
1.1.2. <u>BEGIN..END</u> - Blok-sætninger.....	9
1.1.3. <u>START/END...NEXT...OM</u> - Løkker	10
1.1.4. <u>NOT, AND, OR</u> - Logiske operatører	11
1.1.5. <u>REM, /*</u> - Kommentarer	12
1.1.6. <u>GOTO</u> Gå til label	13
1.1.6.1. <u>ON...GOTO/GOSUB</u> - Betinget hop / subrutinekald.....	14
1.1.7. <u>GOSUB</u> Kald af subrutiner	15
1.1.7.1. <u>RETURN</u> - Returnering fra en subrutine	16
1.2. Felter	17
1.2.1. <u>#xx og kk#xx</u> - Kartoteksfeltter	18
1.2.1.1. <u>#xx(fra,til)</u> - Dele af felter.....	19
1.2.1.2. <u>#xx(nr)</u> - Tabelfelter.....	20
1.2.1.3. Konvertering mellem tal og tekstfeltter	21
1.2.2. <u>SY#xx</u> - Systemfeltter	22
1.2.2.1. <u>#DD, #PD</u> - Dags dato og Pr.dato	23
1.2.2.2. <u>#PP</u> - Sidenummer	24
1.2.2.3. <u>#SN</u> - Systemnavn	25
1.2.2.4. <u>#OK</u> - Resultat af læsning af kartotek	26
1.2.2.5. <u>#UN</u> Brugernavn	27
1.2.2.6. <u>#LIN</u> Linienummer og <u>#LOF</u> Antal linier på siden.....	28
1.2.2.7. <u>#LEVEL</u> - Nuværende total niveau	29
1.2.2.8. <u>kk#RECNO</u> - Sidst anvendte recordnummer fra kartotek kk.....	30
1.2.3. <u>WW#xx</u> - Frifelter (Workfeltter)	31
1.2.3.1. <u>#Dntext</u> - Input data	32
1.2.3.2. <u>#Ptext</u> - Billedfeltter	33
2. Aritmetiske funktioner	34
2.1. <u>ABS</u> - Absolutværdien af et tal.....	35
2.2. <u>FNH</u> - Afrunding til ingen decimaler	36
2.3. <u>FNR</u> - Afrunding til 2 decimaler	37
2.4. <u>FRA</u> - Udskil decimalværdien af et tal	38
2.5. <u>INT</u> - Heltalsværdien af et tal	39
2.6. <u>NOT</u> - Logisk negation	40
2.7. <u>POW</u> - Potensopløftning	41
2.8. <u>RUN</u> - Afrunding til x decimaler.....	42
2.9. <u>RUND</u> - Definition af FNR afrunding	43
2.10. <u>SGN</u> - Undersøg om tal er negativt, nul eller positivt	44
2.11. <u>SQR</u> - Beregn kvadratroden af et tal	45
3. Tekstfunktioner	46
3.1. <u>CONV</u> - Ændring af karakterer i en tekst	47
3.2. <u>EDIT</u> - Editering af et heltal.....	48
3.3. <u>FIND</u> - Find tekst i tekstfelt	49
3.4. <u>LEN</u> - Længden af en tekst.....	50
3.5. <u>LOWER</u> - Konverter tekst til små bogstaver	51
3.6. <u>NAME</u> - Udtræk af for og efternavne.....	52
3.7. <u>NUMBER</u> - Konvertering af 'mystiske' numre	53
3.8. <u>NUMS</u> - Konvertering af tekstfelt til tal.....	54
3.9. <u>PACK</u> - Pakning af et tal.....	55
3.10. <u>SMAA</u> - Konverter tekst til store/små bogstaver - navne	56

3.11. <u>SOGE</u> - Dannelse af søgenøgle ud fra adressefelt	57
3.12. <u>SPOFF</u> - Fjern foran- og bagved-stillede blanke i teksten	58
3.13. <u>UNPACK</u> - Udpakning af et tal	59
3.14. <u>UPPER</u> - Konverter tekst til store bogstaver	60
3.15. <u>USING</u> - Editering af et tal	61
4. Checkcifre og validering	62
4.1. <u>CCODE</u> - Felt checktekst (DATAMASTER checkkodetekst)	63
4.2. <u>CHECK</u> - OCR check	64
4.3. <u>CHEX</u> - Modulus 11 check	65
4.4. <u>VALCH</u> - Check om tekst findes i valideringsstreng	66
4.5. <u>VALID</u> - Check om tal findes i de angivne tilladte værdier	67
5. Dato funktioner	68
5.1. <u>DATE</u> - Dato ÅÅÅÅMMDD	69
5.2. <u>DATECALC</u> - Beregning af en dato	70
5.3. <u>DAY</u> - Beskrivelse af dato på tekstform	71
5.4. <u>FNA</u> - Omregn dato til antal dage fra år 0	72
5.5. <u>FNB</u> - Omregn antal dage fra år 0 til dato	73
5.6. <u>FND</u> - Datovending	74
5.7. <u>FNE</u> - Omregn dato til månedsnummer	75
5.8. <u>FNF</u> - Omregn dato til dagnummer, 360 dage pr.år	76
5.9. <u>FNO</u> - Konverter dato til DDMMÅÅ	77
5.10. <u>FNU</u> - Omregn dato til ugedag	78
5.11. <u>FNV</u> - Omregn dato til ugenr eller ugenr til dato	79
5.12. <u>FNY</u> - Konverter dato til ÅÅÅÅMMDD	80
5.13. <u>MONTH</u> - Beskrivelse af måned på tekstform	81
5.14. <u>TIME</u> - Tid TTMMSS	82
5.15. <u>WDAY</u> - Beskrivelse af ugedag for dato	83
5.16. <u>WEEK</u> - Omregn dato til ugenr eller ugenr til dato	84
5.17. <u>WORKD</u> - Beregn antal arbejdsdage mellem to datoer	85
6. Behandling af flere felter	86
6.1. <u>LET</u> - Beregning af flere felter i et bundt	87
6.1.1. <u>LET</u> - Sæt felter lig værdier i IQ programmer (IQ)	88
6.1.2. <u>LET</u> - Oprettelse af nye kartoteker (RAP)	89
6.2. <u>CLEAR</u> - Nulstil alle felter i et kartotek (RAP)	90
6.3. <u>CLRFLAG</u> - Deaktiver parametre for felter (IQ)	91
6.4. <u>COLOR</u> - Sæt baggrunds box farve for et antal felter	92
6.5. <u>COLORF</u> - Sæt forgrunds tekst farve for et antal felter	93
6.6. <u>DIALOG</u> - Ekstra input dialog	94
6.7. <u>GETFLAG</u> - Aflæs parametre for et felt (IQ)	95
6.8. <u>SETFLAG</u> - Sæt parametre for skærmfelter (IQ)	96
6.9. <u>ZERO</u> - Nulstil et antal felter	97
7. Rapport kontrol	98
7.1. <u>CHAIN</u> - Start af næste rapport eller andet program (RAP)	99
7.1.1. <u>CHAINR</u> - Start program eller extern kommando direkte (RAP)	100
7.1.2. <u>CHAIN</u> - Start IQ program eller extern kommando (IQ)	101
7.2. <u>WAIT</u> - Vent til et program er færdigt (IQ)	102
7.3. <u>COMPILE</u> - Kompiler rapport (RAP)	103
7.4. <u>EXIT</u> - Afslut rapport (RAP)	104
7.4.1. <u>EXIT</u> - Luk et IQ program eller vindue (IQ)	105
7.5. <u>KEYS</u> - Start/Stop angivelser (RAP)	106
7.6. <u>INDEX</u> - Sæt index og start/stop værdi for rapport (RAP)	107
7.7. <u>LTOT</u> - Sæt laveste total niveau (RAP)	108
7.8. <u>MTOT</u> - Sæt det maksimale totalniveau (RAP)	109
7.9. <u>MESS</u> - Skriv meddelelse på skærmen	110
7.10. <u>NOPAS</u> - Intet password/brugernavn på rapporten (RAP)	111

7.11. <u>PAS</u> - Sæt password/brugernavn (RAP).....	112
7.12. <u>PARAMS</u> - Ekstra rapport start parametre (RAP).....	113
7.13. <u>RETURN</u> - Returner fra beregninger	114
7.14. <u>SORTKEY</u> - Frigivelse af extra sorteringsnøgle (RAP).....	115
7.15. <u>SORTWORK</u> - Anvendelse af bestemt sorteringskartotek (RAP)	116
7.16. <u>NÅR</u> - Hvornår der skal beregnes (RAP)	117
8. Printer styring	118
8.1. <u>COPIES</u> - Antal kopier af udskriften (RAP)	119
8.2. <u>PAGE</u> - Skift layout side (RAP)	120
8.3. <u>PRINT</u> - Udskrift af linier fra rapportens layout (RAP).....	121
8.3.1. <u>PRINT</u> - Printer udskrifts kontrol (RAP.).....	122
8.3.2. <u>PRINT(?=</u> - Aflæs printer opsætning (RAP.).....	123
8.4. <u>PRINT(LAB=</u> - Label funktion (RAP).....	124
8.5. <u>PRINTER</u> - Printervalg (RAP.).....	125
8.5.1. <u>PRINTER</u> - Udskrift på flere printere (RAP).....	126
8.6. <u>PRTTOTAL</u> - Manuel styring af totaludskrift (RAP).....	127
8.7. <u>SCRPR</u> T - Genkald gemt skærmpoint (IQ)	128
9. Læsning af kartoteker	129
9.1. <u>READ</u> - Læs en post fra et kartotek	130
9.2. <u>READH</u> - Læs en post og udskriv eventuelt overskrift	131
9.3. <u>READR</u> - Læs record med bestemt recordnummer.....	132
9.4. <u>READX</u> - Læs record med relativ recordnummer angivelse	133
9.5. <u>START</u> - Sæt index og interval for et kartotek.....	134
9.6. <u>NEXT</u> - Hent næste post i interval	135
9.7. <u>OM</u> - Gentag læsning med NEXT	136
9.8. <u>GETKEY</u> - Hent den nuværende nøgle	137
9.9. <u>END</u> - Sæt slut interval efter START	138
9.10. <u>PRIOR</u> - Hent forrige post i interval	139
9.11. <u>SPEED</u> - Optimering af læsestrategi.....	140
10. Skrivning i kartoteker	141
10.1. <u>UPDATE</u> - Tilladelse til skrivning i kartoteker.....	142
10.2. <u>REWRITE</u> - Opdater en eksisterende post i kartoteket	143
10.3. <u>INSERT</u> - Indsæt ny post i kartotek	144
10.4. <u>DELETE</u> - Slet en post i et kartotek.....	145
10.5. <u>WRITE</u> - Opdater eller indsæt post i kartotek	146
11. Export / Import fra eksterne filer.....	147
11.1. <u>EXPORT</u> - Export af data til en tekstfil.....	148
11.2. <u>IMPORT</u> - Import af data fra en tekstfil (RAP)	150
11.2.1. <u>IMPOCONT</u> - Fortsættelse af import (RAP)	151
11.2.2. <u>IMPONEXT</u> - Import af næste record (RAP)	152
11.2.3. <u>IMPOTTHIS</u> - Import af denne record igen (RAP).....	153
11.3. <u>FTP</u> - File Transfer Processor filoverførsler	154
12. Flere firmaer og blanding af kartoteker	155
12.1. <u>ACCESS</u> - Check om en fil findes	156
12.2. <u>COMNO</u> - Firmaid	157
12.3. <u>ENDSUM</u> - Extra grande total ved kørsel med flere hovedkartoteker.	158
12.4. <u>FILENAME</u> - Aktuelt filnavn for et kartotek.....	159
12.5. <u>OPEN</u> - Åbning af en fil med et givet navn	160
12.5.1. <u>OPEN</u> - Temporær lukning af filer	161
12.6. <u>MERGE</u> - Blanding af flere hovedkartoteker på een rapport (RAP).....	162
12.7. <u>OPCOM</u> - Åbning af kartoteker i forskellige firmaer	163
13. DATAMASTER funktioner	164
13.1. <u>DISABLE</u> - Deaktiver input for et program (IQ)	165
13.2. <u>DISP</u> - Visning af ændrede felter (IQ)	166
13.3. <u>DOFUNCTION</u> - Udfør extern funktion (IQ)	167

13.4. <u>ENABLE</u> - Aktiver input for et program (IQ)	168
13.5. <u>FOCUS</u> - Aktiver program (IQ)	169
13.6. <u>FUNC</u> - Bestem hvilken opdatering, der er nødvendig (IQ).....	170
13.7. <u>GETINFO</u> - Hent udvidet program information (IQ/DM)	171
13.8. <u>HELP</u> - Vis meddelelsesbox med hjælp for et felt (IQ)	172
13.9. <u>ISACTIVE</u> - Check om et program er aktivt (IQ)	173
13.10. <u>KEYON</u> - Gem eller vis nøgleinputfeltet (IQ).....	174
13.11. <u>LINE</u> - Hent eller sæt det aktuelle linienummer (IQ/DM).....	175
13.12. <u>LOOP</u> - Kald en rutine for alle records i liniebufferen (IQ)	176
13.13. <u>MENUCH</u> - Flip menu afchecket flag (IQ)	177
13.14. <u>MENUS</u> - Ændring af menuerne (IQ)	178
13.15. <u>MENUUPD</u> - Tilføj til/Kontroller menu (IQ)	179
13.16. <u>NEXTFLD</u> - Spring til dette inputfelt (IQ)	180
13.17. <u>NEXTFLDSEQ</u> - Hop til et inputfelt i en sekvens (IQ)	181
13.18. <u>OBJECTADDSTRING</u> - Adder tekststreng til et objekt (IQ).....	182
13.19. <u>OBJECTCLEAR</u> - Nulstil indholdet af et objekt (IQ)	183
13.20. <u>OBJECTGETSTRING</u> - Hent nummeret på det valgte objekt (IQ/DM)	184
13.21. <u>PLSNEXT</u> - Forbered og læs hovedkartoteket (IQ).....	185
13.22. <u>SEQ</u> - Ændring af inputsekvensen (IQ).....	186
13.23. <u>SETUPD</u> - Marker et kartotek på en linie for opdatering (IQ)	187
13.24. <u>SHOW</u> - Aktiver/Deaktiver/Vis/Gem et felt (IQ/DM)	188
13.25. <u>SUPER</u> - Forbered superindex søgning (IQ).....	189
13.26. <u>TRANSMIT</u> - Opdater andre IQ programmer (IQ)	190
13.27. <u>TRANSSEL</u> - Definer IQ transaktions selektioner (IQ)	191
14. SYSTEM funktioner.....	192
14.1. <u>DEBUG</u> - Aktiver debug vindue (IQ).....	193
14.2. <u>EXEC</u> - Udfør en tekststreng som en beregning	194
14.3. <u>GETFLD</u> - Sæt SY strukture pointers (IQ)	195
14.4. <u>INSTALL</u> - Externe funktioner	196
14.5. <u>SYSPAR</u> - Hent systemparameter	197
14.6. <u>SYSPARSET</u> - Sæt værdien af en systemparameter	198
14.7. <u>USERINFO</u> - Hent brugerinformation fra brugerstyringen.....	199
14.8. <u>WIF</u> - Testprint (IQ)	200
14.9. <u>WIF</u> - Testprint (RAP)	201
14.10. <u>WIFS</u> - Testprint af felter (IQ)	202
Index.....	203

1. Introduktion

SW-Tools produkterne benytter til sit beregningsmodul et BASIC-lignende sprog. Dette muliggør bl.a test af kartotekfelters værdier, matematiske beregninger, behandling af tekst og meget mere.

	Reserverede ord	Synonym	Beskrivelse
Betingede sætninger	IF LET ELSE	HVIS LAD ELLERS	IF udtryk ... IF udtryk LET udtryk ... IF udtryk ... ELSE udtryk
Blok-sætninger	BEGIN END		starter en blok afslutter en blok
Ord benyttet i løkker	BREAK CONTINUE		afbryd løkke gå til slut af løkke og forsæt
Logiske operatører	NOT AND OR	IKKE OG ELLER	forskellig fra 0 og eller
Matematiske operatører	+ - * / %		plus minus gange divider procent beregning
Relations operatører	= > < >= <= <>		lig med større end mindre end større end eller lig med mindre end eller lig med forskellig fra
Kommentarer	REM /*		fuld kommentar linie kommentar efter beregningsudtryk
Hop og subrutiner	GOTO GOSUB RETURN ON..GO..		gå til label: udfør subrutine label: returner fra subrutine betinget hop/rutine kald

Med de ord det BASIC-lignende sprog tilbyder kan mange forskellige beregningsudtryk dannes. Det vil nu kort blive beskrevet med eksempler.

1.1. Eksempler

1.1.1. **IF..ELSE** - Betingede sætninger

Hvis leverandørsaldo (LE#6) større end 1000 kr. så træk 100 kr. fra, ellers læg 47 til.

```
IF LE#6 > 1000 LET LE#6 = LE#6 - 100 ELSE LET LE#6 = LE#6 + 47
```


1.1.2. BEGIN..END - Blok-sætninger

Hvis leverandørsaldo (LE#6) større end 1000 kr. så start blok hvor der trækkes 100 kr. fra saldoen og linie 7 udskrives.

```
IF LE#6 > 1000 THEN BEGIN  
LE#6 = LE#6 - 100  
PRINT(7)  
END
```

Dvs. at alle beregningslinier imellem BEGIN og END kun udføres såfremt betingelsen er opfyldt.

1.1.3. START/END...NEXT...OM - Løkker

Nedenstående løkke læser alle leverandører i intervallet 111-999. Såfremt saldoen er mindre end 1000 behandles leverandøren ikke.

```
START(LE), "111"  
END(LE), "999"  
NEXT(LE)  
IF LE#6 < 1000 CONTINUE  
REM *** behandling af leverandør ***  
OM(LE)
```

Nedenstående løkken læser alle leverandører i intervallet 111-999. Når en saldo større end 10000 mødes stoppes løkken.

```
START(LE), "111"  
END(LE), "999"  
NEXT(LE)  
IF LE#6 > 10000 BREAK /* hvis saldo > 10000 - stop løkken  
OM(LE)  
IF LE#6 > 10000 .....
```

1.1.4. NOT, AND, OR - Logiske operatører

```
IF NOT VA#5 LET VA#5=#DD
```

Såfremt sidste købsdato er 0, sæt da købsdatoen til dags dato. Sætningen svarer til

```
IF VA#5=0 LET VA#5=#DD
```

Såfremt købspris forskellig fra 0 OG sidste købsdato forskellig fra 0, udskriv da linie 5 på rapporten.

```
IF VA#4<>0 AND VA#5<>0 PRINT(5)
```

Såfremt købspris forskellig fra 0 ELLER sidste købsdato forskellig fra 0, udskriv da linie 5 på rapporten.

```
IF VA#4<> OR VA#5<>0 PRINT(5)
```

1.1.5. REM, /* - Kommentarer

```
REM *** denne rapport er udviklet af SW-Tools ApS ***  
REM *** den. 07.09.1997  
IF LE#6 > 1000 LET LE#6 = LE#6 - 100 /* Juster saldo
```

1.1.6. **GOTO** Gå til label

Ved hjælp af GOTO kan man hoppe rundt i beregningerne, typisk afhængig af værdien af et udtryk. En 'label' defineret med 'NAVN:' bestemmer, hvor der hoppes hen. I nedenstående eksempel udskrives linie 7 tre gange.

```
#30 = 0                /* Nulstil tæller
OMIGEN:                /* Label for senere hop
PRINT(7)              /* Gør noget
#30 = #30 + 1         /* Tæl
IF #30 < 3 GOTO OMIGEN /* Gør det 3 gange
```

1.1.6.1. **ON**...GOTO/GOSUB - Betinget hop / subrutinekald

Med ON kan man afhængig af værdien af et felt hoppe til en bestemt label. ON kan anvendes både i forbindelse med GOTO og GOSUB.

```
#30 = 0
ON #7 GOTO ENER,TOER,ENER
#30 = #30 + 1          /* #30 bliver 3 hvis felt 7 ikke er 1,2 eller 3
TOER: #30 = #30 + 1    /* #30 bliver 2 hvis felt 7 er 2
ENER: #30 = #30 + 1    /* #30 bliver 1 hvis felt 7 er 1 eller 3
```

1.1.7. **GOSUB** Kald af subrutiner

Hvis samme beregninger skal udføres flere gange kan man med fordel skrive disse linier som en subrutine, der starter med 'label:' og kaldes med GOSUB

```
#30 = 0          /* Nulstil tæller
GOSUB UD        /* Kald rutinen
GOSUB UD        /* Kald rutinen igen
RETURN         /* Afslut de normale beregninger
UD: #30 = #30 + 1 /* Rutine UD, tæl felt 30 op
PRINT(7)       /* Og udskriv linie 7
RETURN         /* Returner fra subrutinen
```

1.1.7.1. RETURN - Returnering fra en subrutine

Med RETURN afsluttes en subrutine og beregningerne fortsætter fra det sted, hvor man kaldte denne. Se også RETURN funktionen senere, hvor man afslutter beregningerne med en returværdi.

1.2. Felter

1.2.1. #xx og kk#xx - Kartoteksfelter

Man refererer et felt fra et kartotek som:

#xx = feltnr. xx fra hovedkartoteket

kk#xx = feltnr. xx fra kartoteket kk.

Bemærk at kk, KK, Kk, kK refererer til forskellige records fra kartotek KK, man skal normalt anvende små bogstaver kk.

1.2.1.1. #xx(fra,til) - Dele af felter

Dele af felter angives med for eksempel kk#xx(fra,til) og kan anvendes for både tal og tekstfelter.

```
#30 = #2(3,4)          /* Felt 30 bliver karakter 3 og 4 af felt 2
```

For tekstfelter og kun disse kan man desuden sætte en del af feltet lig en tekst:

```
#2="Sorengo og Søn A/S"
```

```
#2(9,14)="xx"        /* Felt 2 bliver "Sorengo xx A/S"
```

1.2.1.2. #xx(nr) - Tabelfelter

Tabelfelter refereres med for eksempel `kk#xx(nr)` hvor `nr` går fra 0 og frem.

Et felt kan være defineret i databasen som et tabelfelt ved i formatet at indeholder for eksempel: `20(003)` angivende 3 extra elementer eller blot være en række sammenhængende felter med samme format, som man ønsker at behandle som en tabel i beregningerne. Bemærk at frifelter også kan defineres som tabelfelter ved hjælp af formatet.

Et eksempel herpå er demo-leverandørkartoteket, hvor navneblokken #2,#3 og #4 også kan opfattes som en tabel #2(0), #2(1) og #2(2)

```
#30 = #2(#31)      /* Frifelt 31 bestemmer navnelinie 0,1 eller 2
PRINT(7)          /* Der herefter udskrives
#2(#31)="xx"      /* Og sættes lig "xx"
```

Bemærk at mystiske værdier kan fremkomme, hvis man overskrider det maksimalt mulige for en tabel og for eksempel anvender #2(4).

1.2.1.3. Konvertering mellem tal og tekstfelter

Man kan blot sætte et talfelt = et tekstfelt, for eksempel #30 = #2, hvorefter man kan regne videre med tallet. Funktionerne NUMBER og NUMS kan anvendes til mere avancerede konverteringer, se disse.

Sættes et tekstfelt = et talfelt, for eksempel #2 = #30, dannes en tekststreng med variabel længde afhængig af tallet i felt 30, for eksempel "123". Normalt vil man anvende #2 = #30 USING "#####" for at angive udseendet af tekstfeltet, se USING funktionen.

1.2.2. SY#xx - Systemfelter

Systemfelter er specialfelter defineret i pseudokartoteket SY der altid vil være til stede. Et par af de mest anvendte systemfelter vil blive nævnt i det følgende, for en komplet oversigt envises til definitionen af SY kartoteket.

Et systemfelt kan enten refereres med nummer (SY#1) eller en forkortelse (#DD) der er angivet først i feltnavnet. Visse af systemfelterne er knyttet til et kartotek og skal refereres med kk#forkortelse som for eksempel kk#RECNO

1.2.2.1. #DD, #PD - Dags dato og Pr.dato

Indtastet ved start af rapporten, (99.99.99).

1.2.2.2. #PP - Sidenummer

Tildeles automatisk fortløbende ved sideskift, (9999).

1.2.2.3. #SN - Systemnavn

Kan anvendes såfremt rapportgeneratoren er installeret med flere systemer, f.eks. forskellige selskaber/kartotekssæt. Bemærk også felterne #SU for subsystem navn og #CN for firmanavn.

1.2.2.4. #OK - Resultat af læsning af kartotek

Lige efter læsning af et kartotek kan man spørge på #OK. Dette felt har værdien 0 hvis en record er læst, andet indikerer fejl.

1.2.2.5. #UN Brugernavn

Man kan anvende #UN for at få det brugernavn for denne PC, der er indtastet i LICENS modulet.

1.2.2.6. #LIN Linienummer og **#LOF** Antal linier på siden

#LIN indeholder den aktuelle printlinier, #LOF antal linier på siden.

1.2.2.7. #LEVEL - Nuværende total niveau

Med #LEVEL kan forskellige beregninger / udskrifter styres afhængig af subtotal niveauet, se RAPGEN brugerhåndbog.

1.2.2.8. kk#RECNO - Sidst anvendte recordnummer fra kartotek kk

Såfremt det anvendte database system er knyttet til recordnumre kan det sidst læste for kartoteket kk findes i kk#RECNO. Bemærk også felterne kk#NUMBER for relativt recordnummer og kk#FILENAME

1.2.3. WW#xx - Frifelter (Workfelter)

Et program vil ved oprettelse få 40 frifelter, der ved første gangs brug skal defineres med et navn og et format og senere kan omdefineres ved dobbeltklik på feltet.

Feltnumrene vil ligge i fortsættelse af hovedkartotekets feltnumre, men felterne blive gemt som WW#1,WW#2,... hvorved en senere udvidelse af hovedkartoteket automatisk vil omnummerere frifelterne.

Antallet af frifelter kan i IQ/DATAMASTER justeres via programparametre, i RAPGEN ved i beregningerne blot at anvende et højere nummer hvorved antallet af frifelter udvides automatisk.

1.2.3.1. #Dntext - Input data

I RAPGEN bevirker et frifelt navn startende med #Dn input af 1 til 7 data ved start af rapporten.

1.2.3.2. #Ptext - Billedefelter

Et frifelt navn startende med #P og defineret som tekstfelt er en reference til et billede.

2. Aritmetiske funktioner

I dette afsnit beskrives forskellige funktioner til talbehandling såsom afrundinger og potensopløftning.

2.1. **ABS** - Absolutværdien af et tal

tal ABS(tal *par1*)

Parametre: *par1* : tal som skal konverteres til en absolut værdi

Beskrivelse: Funktionen returnerer den absolutte værdi af parameteren *par1*. Dvs. den positive værdi uden fortegn.

Returværdi: Den positive værdi af tallet.

Se også: [SGN](#)

Eksempel: #1 = ABS(-123.45) /* Felt #1 vil indeholde værdien 123.45

2.2. **FNH** - Afrunding til ingen decimaler

tal FNH(tal *par1*)

Parametre: *par1* : angiver et tal (med decimaler)

Beskrivelse: Funktionen benyttes til at afrunde et tal med decimaler til et tal uden decimaler.

Returværdi: Tallet uden decimaler.

Se også: FNH, RUN

Eksempel: #1 = FNH(1234.56) /* Felt #1 vil indeholde værdien 1235

2.3. **FNR** - Afrunding til 2 decimaler

tal FNR(tal *par1*)

Parametre: *par1* : angiver et tal (med decimaler)

Beskrivelse: Funktionen benyttes til at afrunde et tal med mere end 2 decimalers værdi til et tal med kun 2 decimaler. Rapportgeneratoren vil altid afrunde resultatet af en beregning til det antal decimaler, som det beregnede felt indeholder. Dog kan man anvende FNR/FNH funktionerne såfremt man ønsker anden form for afrunding.

Afrundingen kan ændres generelt. Ved hjælp af RUND funktionen kan man definere:

Returværdi: Det afrundede tal.

Se også: FNH, RUN, RUND

Eksempel: #1 = FNR(123.456) /* Felt #1 vil indeholde værdien 123.46

2.4. **FRA** - Udskil decimalværdien af et tal

tal FRA(tal *par1*)

Parametre: *par1* : angiver det tal, hvorfra decimalværdien skal returneres

Beskrivelse: Funktionen udskiller decimalværdien af et tal og returnerer denne.

Returværdi: Decimalværdien som 0.<decimalværdi>.

Se også: [FNH](#), [FNR](#), [RUN](#)

Eksempel: #1=FRA(123.456) /* Giver 0.456 , #1=FRA(-12.345) /* Giver -0.345

2.5. **INT** - Heltalsværdien af et tal

tal INT(tal *par1*)

Parametre: *par1* : angiver et tal

Beskrivelse: Funktionen returnerer heltalsværdien af et tal, dvs. nærmeste lavere værdi uden decimaler.

Returværdi: Heltalsværdien.

Se også: [FRA](#)

Eksempel: #1=INT(1234.56) /* Giver 1234 , #1=INT(-12.345) /* Giver -13

2.6. **NOT** - Logisk negation

tal NOT(tal *par1*)

Parametre: *par1* : angiver et tal

Beskrivelse: Funktionen returnerer 1 hvis *par1* er nul, 0 hvis *par1* ikke er nul.

Returværdi: 0 eller 1.

Se også: [SGN](#)

Eksempel: NOT(1) /* er 0

2.7. **POW** - Potensopløftning

tal POW(tal *par1*, tal *par2*)

par2 : angiver eksponenten

Beskrivelse: Funktionen opløfter tallet *par1* til den *par2* potens.

Returværdi: Potensen.

Se også: [SQR](#)

Eksempel: #1=POW(8,3) /* Giver 512 (8*8*8) , #1=POW(4,0.5) /* Giver 2

2.8. **RUN** - Afrunding til x decimaler

tal RUN(tal *par1*, tal *par2*)

par1 : Antal decimaler, der skal afrundes til

Beskrivelse: RUN funktionen afrunder det angivne tal til det angivne antal decimaler.

Returværdi: Det afrundede tal

Se også: [FNH](#), [FNR](#), [INT](#)

Eksempel: #1=RUN(-123.4567,3) /* Felt 1 vil antage værdien -123.457

2.9. **RUND** - Definition af FNR afrunding

tal RUND(tal *par1*, tal *par2*)

par2 : Det antal decimaler, der skal afrundes TIL, fx. 2

Beskrivelse: RUND funktionen definerer, hvordan FNR funktionen skal afrunde.

Hvis *par1* er positiv rundes OP, hvis *par1* er negativ rundes NED.

Returværdi: Ingen.

Se også: [FNR](#)

Eksempel:

```
RUND(-25,2)    /* Der rundes NED til nærmeste 25 øre med 2 decimaler
RUND(5,2)      /* Der rundes OP til nærmeste 5 øre
RUND(1,3)      /* Der afrundes til 3 decimaler
RUND(1,2)      /* FNR funktionen fungerer som normalt
```

2.10. **SGN** - Undersøg om tal er negativt, nul eller positivt

tal `SGN(tal par1)`

Parametre: `par1` : angiver et tal

Beskrivelse: Funktionen undersøger om tallet er negativt, nul eller positivt.

Returværdi:

-1	Tallet er negativt
0	Tallet er nul
1	Tallet er positivt

Se også: INT, NOT

Eksempel: `#1=SGN(-123.45) /* Felt #1 vil antage værdien -1.`

2.11. **SQR** - Beregn kvadratroden af et tal

tal SQR(tal *par1*)

Parametre: *par1* : angiver det tal, som kvadratroden skal tages af

Beskrivelse: Funktionen beregner kvadratroden af tallet i *par1*.

Returværdi: Kvadratroden.

Se også: [POW](#)

Eksempel: #1=SQR(4) /* Giver 2

3. Tekstfunktioner

Dette afsnit beskriver funktioner for generel ændring af tekster samt editering af tal til tekst.

3.1. **CONV** - Ændring af karakterer i en tekst

tekst CONV(tekst *par1*, tekst *par2*, tekst *par3*)

par3 : angiver de nye karakterer som skal indsættes

Beskrivelse: Funktionen checker hver karakter i teksten *par1*. Såfremt karakteren svarer til en af de i *par2* angivne, udskiftes denne med den nye karakter i *par3*. For at dette skal fungere korrekt skal parameter 2-3 indeholde lige mange karakterer. Dvs. hvis parameter 1 indeholder "abc" og parameter 2 teksten "ABC", vil funktionen udskifte a med A, b med B og c med C.

Returværdi: Teksten, hvor de ønskede karakterer er konverteret.

Se også: LOWER, SMAA, UPPER

Eksempel: #1 = CONV("hans", "hn", "lr") /* Giver "lars"

3.2. **EDIT** - Editering af et heltal

tekst EDIT(tal *par1*, tekst *par2*)

par2 : USING maske for editering

Beskrivelse: Edit funktionen konverterer et heltal til et tekstfelt, idet den givne USING maske bestemmer tekstens udseende.

Returværdi: Den editerede tekst.

Se også: NUMBER, USING

Eksempel:

```
#1 = EDIT(-123, "&&&, &&")           /* Giver "001,23"
#1 = EDIT(123, "##&-#&&&")          /* Giver " 0- 123"
#1 = EDIT(123, "fx.# og ##")       /* Giver "fx.1 og 23"
```


3.3. **FIND** - Find tekst i tekstfelt

tal FIND(*tekst par1*, *tekst par2*, *tal par3*, *tal par4*, *tal par5*)

Beskrivelse: Funktionen søger efter teksten *par1* i teksten *par2*. Begge parametre skal angives i "" (gåseøjne).

Returværdi: Returnerer -1 hvis teksten ikke er fundet, ellers et positivt tal svarende til positionen i den tekst der er søgt i - startende fra 1.

Se også:

Eksempel:

```
#1 = "Dette er en tekst"
#2 = FIND("en", #1)           /* Felt #2 vil indeholde værdien 10.
```

3.4. LEN - Længden af en tekst

tal LEN(*tekst par1*)

Parametre: *par1* : angiver en tekst

Beskrivelse: Funktionen udregner længden af en tekst.

Returværdi: Længden af teksten.

Se også: SPOFF

Eksempel:

```
#1 = "SW-Tools ApS"
```

```
#2 = LEN(#1) /* returner længden af en tekst
```

Felt #2 vil antage værdien 12, eftersom der er 12 karakterer i #1.

3.5. LOWER - Konverter tekst til små bogstaver

tekst LOWER(tekst *par1*)

Parametre: *par1* : angiver en tekst som skal konverteres

Beskrivelse: Funktionen konverterer en tekst til små bogstaver, dvs. alle bogstaver A-Z, Æ, Ø og Å konverteres til a-z, æ, ø og å.

Returværdi: Den konverterede tekst.

Se også: CONV, SMAA, UPPER

Eksempel:

```
#1 = "DETTE er en TEST ÆØÅ"
```

```
#2 = LOWER(#1) /* Felt #2 vil antage teksten "dette er en test æøå"
```

3.6. **NAME** - Udtræk af for og efternavne

tekst NAME(tekst *par1*, tal *par2*)

Beskrivelse: Funktionen udtrækker for så vidt muligt fornavn og efternavn fra det givne tekstfelt og returnerer disse ifølge *par2*. Denne værdien kan for eksempel anvendes til sortering.

Til dette anvendes SSV tekstfilen WORDS.DAN, hvor hver linie indeholder et specialord som Hr, Fru, ApS samt en eventuel erstatning (Herr;Hr.) for disse.

Returværdi: Navn ifølge *par2*.

Se også: SMAA, SOGE

Eksempel:

```
#1 = NAME("HR CHRIS HANSON",0) /* Giver "HANSON, CHRIS Hr."
#1 = NAME("OLSEN, MICHAEL",1) /* Giver "MICHAEL OLSEN"
```

3.7. NUMBER - Konvertering af 'mystiske' numre

tal NUMBER(*tekst par1*)

Parametre: *par1* : En tekst, der indeholder et tal

Beskrivelse: NUMBER funktionen plukker et tal ud af et tekstfelt, selv om der måtte være karakterer imellem cifrene.

Returværdi: Det fundne heltal, der dannes aldrig decimaler.

Se også: EDIT, NUMS, USING

Eksempel:

```
#1=NUMBER("33)33 05 56") /* Telefonnummer omdannes til tal 33330556  
#1=NUMBER("31/03-1997") /* Dato omdannes til talværdi 31031997  
#1=NUMBER("ab1cd2&3.4") /* Giver 1234
```

3.8. **NUMS** - Konvertering af tekstfelt til tal

tal NUMS(*tekst par1*)

Parametre: *par1* : Tekstfelt indeholdende et tal

Beskrivelse: I en linie som #1=#2, hvor #1 er numerisk og #2 er et tekstfelt, konverteres et eventuelt tal i #2 automatisk fra tekst til tal. Samme resultat kunne være opnået med #1=NUMS(#2), men NUMS er ikke tvungen.

Ønsker man at regne med værdien af tekstfelter, som #1=#2+#3, er det nødvendigt at bruge #1=NUMS(#2)+NUMS(#3) for at angive, at teksterne skal konverteres til tal først.

Returværdi: Talværdien af tekstfeltet. Decimalkomma skal være angivet som punktum.

Se også: NUMBER

Eksempel: #1 = NUMS("ab111") + NUMS("222,22 test") + NUMS("333.33")

Felt 1 bliver summen af tallene i tekstfelterne = 555.33

3.9. **PACK** - Pakning af et tal

tekst PACK(tekst *par1*, tal *par2*)

par2 : 0, anvendes ikke, reserveret for fremtidig paktype

Beskrivelse: Modsvare 8870 basic call 60,A\$,B\$ som B\$=PACK(A\$)

Returværdi: Den pakkede værdi af feltet.

Se også: UNPACK

Eksempel: #1 = PACK(#2) /* #1 bliver #2 pakket.

3.10. SMAA - Konverterer tekst til store/små bogstaver - navne

tekst SMAA(tekst *par1*)

Parametre: *par1* : angiver den tekst som skal konverteres

Beskrivelse: Funktionen konverterer teksten i *par1* til store og små bogstaver. Dvs. det første bogstav i hvert ord vil blive stort, mens resten konverteres til små bogstaver. SSV tekstfilen WORDS.DAN checkes for første/sidste ord, fx. Hr. eller ApS. Findes dette anvendes den her angivne bogstavering/erstatning.

Bemærk at SMAA rutinen kan anvendes i DATAMASTER også til online konvertering af navnefelter.

Returværdi: Den konverterede tekst.

Se også: CONV, LOWER, NAME, UPPER

Eksempel:

```
#1 = SMAA("MICHAEL OLSEN") /* Giver "Michael Olsen"  
#1 = SMAA("SORENCO GMBH") /* Giver "Sorencø GmbH"
```


3.11. **SOGE** - Dannelse af søgenøgle ud fra adressefelt

tekst SOGE(tekst *par1*, tal *par2*)

par2 : Længden af navnedelen i resultatet.

Beskrivelse: Ud fra den givne adresse isoleres gadenavnet og gadenummeret. Disse sammenstilles herefter til et felt med gadenavnet med en fast længde ifølge *par2* efterfulgt af gadenummeret. Dette felt kan for eksempel anvendes til sortering eller søgning.

Returværdi: Gadenavn med længden *par2* efterfulgt af 4-cifret gadenummer.

Se også: LOWER, NAME, SMAA, UPPER

Eksempel:

```
#1 = SOGE("Duevej 3",10)      /* Giver "Duevej_____3"  
#1 = SOGE("27, Rue de Saute",8) /* Giver "RuedeSau__27"
```

3.12. **SPOFF** - Fjern foran- og bagved-stillede blanke i teksten

tekst SPOFF(tekst *par1*, Bitflag *par2*)

Beskrivelse: Funktionen fjerner alle foran- og bagved-stillede blanke i teksten. Ydermere reduceres alle blanke steder i teksten til kun en blank karakter.

Returværdi: Den konverterede tekst.

Se også: [LEN](#)

Eksempel:

```
#1=" Dette er en tekst "  
#2=SPOFF(#1) /* Felt #2 vil antage værdien "Dette er en tekst".
```

3.13. UNPACK - Udpakning af et tal

tekst UNPACK(tekst *par1*, tal *par2*)

par2 : 0, anvendes ikke, reserveret for fremtidig paktype

Beskrivelse: Modsvarende 8870 basic call 61,A\$,B\$ som B\$=UNPACK(A\$)

Returværdi: Den udpakkede værdi af feltet.

Se også: PACK

Eksempel: #1=UNPACK(#2) /* #1 bliver #2 udpakket.

3.14. UPPER - Konverter tekst til store bogstaver

tekst UPPER(tekst *par1*)

Parametre: *par1* : angiver en tekst som skal konverteres

Beskrivelse: Funktionen konverterer en tekst til store bogstaver, dvs. alle bogstaver a-z, æ, ø og å konverteres til A-Z, Æ, Ø og Å.

Returværdi: Den konverterede tekst.

Se også: LOWER, SMAA

Eksempel:

```
#1="Dette er en test"
```

```
#2=UPPER(#1) /* Felt #2 vil antage teksten "DETTE ER EN TEST"
```

3.15. USING - Editering af et tal

tekst USING(tal *par1*, tekst *par2*)

par1 : USING maske for editering

Beskrivelse: USING funktionen konverterer et tal til et tekstfelt, idet den givne USING maske bestemmer tekstens udseende.

Funktionen kan ikke alene anvendes som normal funktion, men også med syntaxen: tekstfelt = tal USING "maske"

Returværdi: Den editerede tekst.

Se også: [EDIT](#)

Eksempel:

```
#1 = USING(-123, "&&&, &&")           /* Giver "001,23"
#1 = USING(123.45, "#####")        /* Giver "_123"
#1 = USING(1234.56, "###,###.##")   /* Giver "_1,234,56"
#1 = 123.45 USING "#####"         /* Giver "_123"
```

4. Checkcifre og validering

Dette afsnit beskriver funktionerne for checkcifre beregning samt validering af tekst og tal.

4.1. **CCODE** - Felt checktekst (DATAMASTER checkkodetekst)

tekst CCODE(tekst *par1*, felt *par2*)

par2 : Det felt, der indeholder checkdefinitionen, fx. "7", "#7", "va#7", "va07"

Beskrivelse: Denne funktion slår op i Data Dictionaryet for det angivne felt *par2* og finder checkdefinitionerne herfor. Teksten knyttet til værdien *par1* returneres.

Returværdi: Checkteksten. Blank angiver ikke tilladt, "-" at ingen check er defineret.

Se også: VALID, VALCH

Eksempel: #1 = CCODE(9,"va#7") /* Giver "Special"

4.2. CHECK - OCR check

tekst CHECK(tekst *par1*)

Parametre: *par1* : angiver er nr, f.eks et kundenr

Beskrivelse: Funktionen behandler et nr og returnerer en tekst indeholdende et OCR checknr, der herefter kan udskrives på et girokort.

#47=CHECK (#19) vil bevirke at OCR checkcifferet modulus 10 med vægtene 212121.. beregnes for tekstfeltet #19 og påsættes bagerst.

CHECK("123456789012345") vil altså returnere en tekst med een karakter længere: "1234567890123452".

Returværdi: Teksten plus et OCR checkciffer.

Se også: CHEX

Eksempel: #1 = CHECK("33330556") /* Giver "333305563"

4.3. **CHEX** - Modulus 11 check

tekst CHEX(tekst *par1*, tekst *par1*)

par2 : Vægte for beregning af checkciffer, 2 cifre pr. input karakter

Beskrivelse: #47 = CHEX (#15, "01020304") vil på samme måde som CHECK rutinen beregne et checkciffer og påsætte dette på tekstfeltet. Checkcifferet beregner modulus 11 med vægtene 01, 02, 03, 04 iflg. anden parameter, hver 2 cifre heri svarer til vægten for 1 ciffer i feltet.

Returværdi: Teksten plus checkciffer.

Se også: [CHECK](#)

Eksempel: #2=CHEX("330556", "010203040506") /* Giver "3305569"

4.4. **VALCH** - Check om tekst findes i valideringsstreng

tal VALCH(*tekst par1*, *tekst par2*)

par2 : angiver de tilladte værdier adskilt med komma

Beskrivelse: Funktionen checker om *par1* findes blandt de værdier angivet i *par2*. Alle værdier som angives i *par2* skal adskilles med , (komma).

Returværdi: Returnerer 0 hvis *par1* ikke findes i *par2*.

Se også: CCODE, VALID

Eksempel: #1=VALCH("Chris", "Anne,Chris,Ole,Michael") /* Felt #1 vil antage værdien 2.

4.5. **VALID** - Check om tal findes i de angivne tilladte værdier

tal VALID(tal *par1*, tal *par2*, tal *par3*)

. **Beskrivelse:** Funktionen checker om værdien i *par1* er tilladt ved at checke de angivne værdier i *par2*. Syntaksen for de tilladte værdier er som følger:

"1,2,8-10,12" Dvs. værdierne 1, 2, 8 til 10 og 12 er tilladte.

"-1,2,8-10,12" Angives minus forrest er de følgende værdier IKKE tilladt.

#20="1-3,8-12"

VALID(15, #20, 1)

vil ændre intervallet i textfeltet #20 ved at indsætte 15 således at #20 bliver: "1-3,8-12,15"

Returværdi: Returnerer 0 hvis *par1* ikke findes i *par2*.

Se også: CCODE, VALCH

Eksempel: #1 = VALID(9, "1,2,8-10,12")

Felt #1 vil antage værdien 3 svarende til værdien blev fundet i tredje interval.

5. Dato funktioner

Datoberegning er en selvstændig videnskab, der beskrives i dette kapitel.

5.1. DATE - Dato ÅÅÅÅMMDD

tal DATE()

Returværdi: Den aktuelle dato på formen ÅÅÅÅMMDD.

5.2. **DATECALC** - Beregning af en dato

Date DATECALC(Date *par1*, tal *par2*, tal *par3*, tal *par4*, tal *par5*)

par5 : angiver dag(e) DD

Beskrivelse: Funktionen kan benyttes til at sætte en dato, addere eller trække fra en dato. Hvis *par2* sættes til 0 kan datoen sættes ved brug af parametrene *par3-par5*. Såfremt parameter 3, 4 og 5 er sat, vil parameter 1 ikke blive behandlet. Ønsker man derimod kun at sætte måneden tager funktionen udgangspunkt i datoen angivet i *par1* og ændrer måneden til den i *par4*.

Returværdi: Den beregnede dato på formen ÅÅÅMMDD.

Se også: DAY, FNA, FNB, FND, FNU, FNV, FNY, MONTH, WDAY, WORKD

Eksempel:

```
#1=DATECALC(0, 0, 1997, 10, 16) /* sæt datoen 16.oktober 1997 (19931016)
#1=DATECALC(19970101, 1, 0, 2, 0) /* læg 2. måneder til datoen (19970301)
#1=DATECALC(19971016, 2, 1, 2, 3) /* træk 1 år, 2 måneder og 3 dage
                                fra datoen (19960813)
```

5.3. **DAY** - Beskrivelse af dato på tekstform

tekst DAY(Dato *par1*)

Parametre: *par1* : angiver en dato på formen ÅÅÅMMDD

Beskrivelse: Funktionen danner en tekststreng, der beskriver datoen som: <?> <ugedag> den. <dag> <månedsbetegnelse> <år>

Såfremt dagen er en helligdag vil det første tegn <?> være en *, hvis kun halv helligdag en /, ellers blankt. Der anvendes samme kalender som beskrevet for WORKD.

Returværdi: Tekststreng med datoen.

Se også: DATECALC, FNA, FNB, FND, FNU, FNV, MONTH, WDAY, WORKD

Eksempel: #1 = DAY(19931016) /* dan tekststreng for 16. oktober 1993

Felt #1 vil indeholde værdien "* Lørdag d. 16 oktober 1993"

5.4. **FNA** - Omregn dato til antal dage fra år 0

tal FNA(Dato *par1*, tal *par2*)

Beskrivelse: Funktionen omregner den angivne dato til det antal dage som er gået siden år 0. Funktionen kan bl.a benyttes til at udregne forskellen mellem to datoer (i dage.).

Returværdi: Antal dage fra år 0.

Se også: FNB, FND, FNU, FNV, DATECALC, DAY, MONTH, WDAY, WORKD

Eksempel:

```
#1 = 19931215          /* datoen 15. december 1993
#2 = FNA(#1)          /* hvormange dage siden år 0 ?
#3 = #2 - FNA(19931202) /* hvormange dage siden 2. december ?
```

Felt #2 vil indeholde værdien 728277 og felt #3 værdien 13

5.5. **FNB** - Omregn antal dage fra år 0 til dato

Dato FNB(tal *par1*, tal *par2*)

Beskrivelse: Funktionen omregner antal dage fra år 0 til en dato ÅÅÅMMDD. Dvs. at et tal returneret fra funktionen FNA() kan gives som parameter til denne funktion og hermed returnere en korrekt dato.

Returværdi: Returværdien er en dato på formen ÅÅÅMMDD.

Se også: DATECALC, DAY, FNA, FND, FNU, FNV, MONTH, WDAY, WORKD

Eksempel:

```
#1 = FNA(19931215) /* omregn dato 15. december 1993
```

```
#2 = FNB(#1 + 9) /* læg 9 dage til og omregn til dato ÅÅÅMMDD
```

Felt #2 vil indeholde værdien 19931224, dvs. 24. december 1993

5.6. FND - Datovending

Dato FND(Dato *par1*)

Parametre: *par1* : angiver en dato DDMMÅÅ

Beskrivelse: Denne funktion er relevant i systemer, hvor datoen lagres som kun 6 cifre. Bruges til vending af datoer, enten for at få et pænt udskriftsformat, eller for at få datoen til at passe til selektion, sortering eller en af de andre datofunktioner.

Specielt ved selektioner er man nødt til sørge for, at datoerne findes på formen ÅÅMMDD, idet maskinen betragter datoen som et helt almindeligt tal, hvor f.eks.:

970101 er større end 961231 men 311296 er større end 10197

man er således nødt til at anvende FND(#7) i stedet for #7 hvis felt 7 er et datofelt på formen DDMMÅÅ.

Returværdi: Returværdien er en dato på formen ÅÅMMDD eller DDMMÅÅ.

Se også: DATECALC, DAY, FNA, FNO, FNU, FNV, FNY, MONTH, WDAY, WORKD

Eksempel:

#1 = FND(310395)	/* Giver 950331
#1 = FND(950331)	/* Giver 310395
#1 = FND(19950331)	/* Giver 310395

5.7. **FNE** - Omregn dato til månedsnummer

tal FNE(Dato *par1*)

Parametre: *par1* : angiver en dato på formen ÅÅÅMMDD eller ÅÅMMDD

Beskrivelse: Funktionen kan benyttes til beregning af månedsintervaller.

Returværdi: Returværdien er Året*12 + Måned (ÅÅ*12+MM)

Se også: DATECALC, DAY, FNA, FNB, FND, FNV, MONTH, WDAY, WORKD

Eksempel: #1 = FNE(19950331) /* giver 1143 = 95*12+3

5.8. **FNF** - Omregn dato til dagnummer, 360 dage pr.år

tal FNF(Dato *par1*)

Parametre: *par1* : angiver en dato på formen ÅÅÅMMDD eller ÅÅMMDD

Beskrivelse: Funktionen omregner datoen til et antal dage fra år 0, idet der regnes med 360 dage pr. år. Det samme som FNA(dato,360)

Returværdi: Antal dage fra år 0.

Se også: [FNA](#)

Eksempel:

#1 = FNF(19950331) /* giver 718290

#1 = FNF(950331) /* giver 34290

5.9. FNO - Konverter dato til DDMMÅÅ

Dato FNO(Dato *par1*)

Parametre: *par1* : Dato på formen DDMMÅÅ, ÅÅMMDD eller ÅÅÅÅMMDD

Beskrivelse: Uanset hvordan datoer venter i inputfeltet returneres DDMMÅÅ, der herefter kan anvendes til udskrift.

Returværdi: DDMMÅÅ

Se også: FND, FNY

Eksempel:

```
#1 = FNO(310395)          /* Giver 310395
#1 = FNO(950331)          /* Giver 310395
#1 = FNO(19950331)       /* Giver 310395
```

5.10. **FNU** - Omregn dato til ugedag

tal FNU(Dato *par1*)

Parametre: *par1* : angiver en dato på formen ÅÅÅMMDD

Beskrivelse: Funktionen benyttes til at fastslå hvilken ugedag en given dato er.

Se også: DATECALC, DAY, FNA, FNB, FND, FNV, MONTH, WDAY, WORKD

Eksempel: #1 = FNU(19931215) /* hvilken dag er 15. december 1993 ?

Felt #1 vil indeholde værdien 4 (=Onsdag)

5.11. **FNV** - Omregn dato til ugenr eller ugenr til dato

tal FNV(tal *par1*)

Parametre: *par1* : angiver en dato på formen ÅÅÅÅMMDD, eller et ugenr som ÅÅÅÅUU

Beskrivelse: Funktionen konverterer en dato til et ugenr ÅÅÅÅUU, såfremt *par1* er en dato. Hvis *par1* derimod er et ugenr ÅÅÅÅUU vil funktionen returnere en dato svarende til den sidste søndag før den angivne uge. Samme som WEEK(dato)

Returværdi: Returnerer et tal ÅÅÅÅUU, hvor ÅÅÅÅ = årstal og UU = ugenr, eller en dato ÅÅÅÅMMDD.

Se også: DATECALC, DAY, FNA, FNB, FND, FNU, MONTH, WDAY, WEEK , WORKD

Eksempel:

```
#1 = FNV(19931016) /* udregn ugenr for den 16. oktober 1993
```

```
#2 = FNV(#1) /* udregn sidste søndag før ugenr 41
```

Felt #1 vil antage værdien 199341, svarende til ugenr 41. Felt #2 vil indeholde datoen 19931010.

5.12. FNY - Konverter dato til ÅÅÅÅMMDD

Dato FNY(Dato *par1*)

Parametre: *par1* : Dato på formen DDMMÅÅ, ÅÅMMDD eller ÅÅÅÅMMDD

Beskrivelse: Uanset hvordan datoer vender i inputfeltet returneres ÅÅÅÅMMDD, der herefter kan anvendes i de videre beregninger.

Returværdi: ÅÅÅÅMMDD

Se også: FND, FNO

Eksempel:

```
#1 = FNY(310395)           /* Giver 19950331
#1 = FNY(950331)         /* Giver 19950331
#1 = FNY(19950331)       /* Giver 19950331
```


5.13. **MONTH** - Beskrivelse af måned på tekstform

tekst MONTH(Dato *par1*)

Parametre: *par1* : angiver en dato på formen ÅÅÅMMDD

Beskrivelse: Funktionen danner en tekst svarende til navnet på den ønskede måned.

Returværdi: Returnerer navnet på måneden.

Se også: DATECALC, DAY, FNA, FNB, FND, FNU, FNV, WDAY, WORKD

Eksempel: #1 = MONTH(19931016) /* dato 16. oktober 1993

Felt #1 vil antage værdien "oktober".

5.14. **TIME** - Tid TTMMSS

tal TIME()

Returværdi: Den aktuelle tid på formen TTMMSS.

5.15. **WDAY** - Beskrivelse af ugedag for dato

tekst WDAY(Dato *par1*)

Parametre: *par1* : angiver en dato på formen ÅÅÅMMDD

Beskrivelse: Funktionen danner en tekststreng, der beskriver datoen som: <?> ugedag
Såfremt dagen er en helligdag vil det første tegn <?> være en *, hvis kun halv helligdag en /,
ellers blankt. Der anvendes samme kalender som beskrevet for WORKD.

Returværdi: En tekststeng indeholdende dagen.

Se også: DATECALC, FNA, FNB, FND, FNU, FNV, MONTH, WDAY, WORKD

Eksempel: #1 = WDAY(19931016) /* Felt #1 vil indeholde værdien "*Lørdag"

5.16. WEEK - Omregn dato til ugenr eller ugenr til dato

tal WEEK(tal *par1*)

Parametre: *par1* : angiver en dato på formen ÅÅÅÅMMDD, eller et ugenr som ÅÅÅÅUU

Beskrivelse: Funktionen konverterer en dato til et ugenr ÅÅÅÅUU, såfremt *par1* er en dato. Hvis *par1* derimod er et ugenr ÅÅÅÅUU vil funktionen returnere en dato svarende til den sidste søndag før den angivne uge. Samme som FNV(dato).

Returværdi: Returnerer et tal ÅÅÅÅUU, hvor ÅÅÅÅ = årstal og UU = ugenr, eller en dato ÅÅÅÅMMDD.

Se også: FNV

Eksempel:

```
#1 = WEEK(19931016) /* udregn ugenr for den 16. oktober 1993
```

```
#2 = WEEK(#1) /* udregn sidste søndag før ugenr 41
```

Felt #1 vil antage værdien 199341, svarende til ugenr 41. Felt #2 vil indeholde datoen 19931010.

5.17. WORKD - Beregn antal arbejdsdage mellem to datoer

tal WORKD(Dato *par1*, Dato *par2*)

par2 : angiver en dato på formen ÅÅÅÅMMDD

Beskrivelse: Funktionen udregner antal arbejdsdage mellem to datoer.

#47 = WORKD (#15, #PD) vil beregne antallet af faktiske arbejdsdage fra datoen i felt 15 til den indtastede pr. dato.

Rutinen beregner først antallet af dage mellem de to datoer, hvorefter alle lørdage og søndage fratrækkes. Herefter slås der op i en indbygget kalender, hvor samtlige forskydelige og eventuelt halve helligdage, der ikke er lørdage eller søndage, er angivet, og disse fratrækkes, såfremt de ligger i det givne interval.

Denne indbyggede kalender kan om nødvendigt tilpasses individuelt. Funktionen baserer sin beregning på filen RAPDAY.DAN. Denne fil er en SSV tekstfil, hvor hver linie indeholder en helligdag på formen ÅÅÅÅMMDD. Såfremt datoen angiver en halv helligdag følger angives for eksempel 19960605;50 idet det andet felt giver procenten.

Returværdi: Returnerer antallet af arbejdsdage mellem to datoer.

Se også: DATECALC, FNA, FNB, FND, FNU, FNV, MONTH, WDAY, WORKD

Eksempel: #1 = WORKD(19930420, 19930430) /* Felt #1 vil antage værdien 19.

6. Behandling af flere felter

I dette afsnit beskrives behandling af bundter af felter, specielt LET funktionen.

6.1. **LET** - Beregning af flere felter i et bundt

tal LET(felter *par1*)

Parametre: *par1* : angiver et eller flere felter

Beskrivelse: Funktionen benyttes til at behandle op til flere felter på en gang. Felterne kan beregnes med følgende udtryk: Felter **XX** konstant/felt, hvor **xx** kan være

Operator	Beskrivelse
=	sæt felterne lig med
+=	adder til felterne
-=	træk fra felterne
*=	gang felterne med
/=	divider felterne med
%=	sæt felterne lig med divisionsresten
&=	udfør logisk AND operation på felter
=	udfør logisk OR operation på felter

Returværdi: Returnerer 0 hvis beregning er foretaget korrekt.

Se også: CLEAR, ZERO

Eksempel:

Letudtryk

LET("#1-10=12")

LET("#20,25=3,7")

LET("#20-25=le#1-10")

LET("#20-25=le#1-2")

LET("le#1,3,va#7=#1,ku#3")

LET("#20-25+=1")

Funktion

Felt 1 til 10 sættes lig 12

#20=3 og #25=7

Felt 20-25 sættes lig kartoteket le felt 1-6

#20=#22=#24=le#1, #21=#23=#25=le#2

Flere kartoteker kan indgå

Der lægges 1 til alle felterne 20-25

6.1.1. **LET** - Sæt felter lig værdier i IQ programmer (IQ)

tal LET(felter *par1*)

Parametre: *par1* : angiver et eller flere felter

Beskrivelse: LET funktionen er udvidet til også at kunne arbejde mellem flere programmer og meller linier i et transaktionsprogram.

Returværdi: Returnerer 0 hvis beregning er foretaget korrekt.

Se også:

Eksempel:

Letudtryk

LET (20.#1-3=#1-3)

LET (#1-3=20.#4-6)

LET (#10=#3.4)

Funktion

Sætter felt 1-3 for program 20 = dette programs #1-3

Sætter felt 1-3 i dette program til #4-6 fra program 20

Sætter felt 10 lig med felt 3 fra linie 4

6.1.2. LET - Oprettelse af nye kartoteker (RAP)

tal LET(felter *par1*)

Parametre: *par1* : angiver et eller flere felter

Beskrivelse: LET funktionen kan anvendes til af bygge nye kartoteker.

Returværdi: Returnerer 0 hvis beregning er foretaget korrekt.

Se også: INSERT, UPDATE, *Rapgen Manual*

Eksempel:

Letudtryk

LET (aa=#1-3,87,le#2)

LET (aa=#1-3,6K,15D)

LET (aa=#1-3,6,15:2,NP)

LET (aa=#1-3),12000

LET (aa=#1-3),-1

LET (aa=#1-3),1000,xnet

LET (aa=#1-3) -acc

LET (07/aa=#1-3),25

Funktion

Definer fil aa, nøgle=aa#1, type=1.database driver

Nøgler aa#4 og aa#5 (Duplikater)

Nøgler aa#2 og rel.recnr (Duplikater)

12000 records (standard er 1000 hvis relevant)

Filen skal bygges ved hver start

Filen er en XNET fil

Filen er en Access fil, bygges hver gang

Lu kan angives for Basic filer

6.2. **CLEAR** - Nulstil alle felter i et kartotek (RAP)

tal CLEAR(kartotek *par1*)

Parametre: *par1* : angiver forkortelsen på kartoteket

Beskrivelse: Funktionen nulstiller alle felter for et kartotek.

Returværdi: Returnerer 0 hvis ok.

Se også: ZERO

Eksempel:

```
UPDATE (1)           /* rapporten opdaterer i kartotek
CLEAR (VA)           /* nulstil alle felter fra varekartoteket
VA#1 = "1234"        /* varenummer
INSERT (VA)          /* indsæt ny post i varekartoteket
```

Eksemplet indsætter en ny post i varekartoteket. Grundet funktionen CLEAR() er alle andre felter end varenr nulstillet.

6.3. CLRFLAG - Deaktiver parametre for felter (IQ)

CLRFLAG(*felte par1*, *tal par2*, *tal par3*)

Beskrivelse: Hvert felt på skærmen er tilknyttet et antal parametre (bits), der definerer, hvordan feltet anvendes. SETFLAG funktionen kan anvendes til at sætte disse parametre, CLRFLAG til at deaktivere dem.

Se også: SETFLAG, GETFLAG

Eksempel: CLRFLAG("#12,44",7,0)

6.4. **COLOR** - Sæt baggrunds box farve for et antal felter

COLOR(felter *par1*, FarveRød *par2*, FarveGrøn *par3*, FarveBlå *par4*)

par4 : Blå farveværdi (0-255)

Beskrivelse: Baggrundsfarven for de angivne felter sættes til RGB værdien, dvs. feltets box udfyldes med den angivne farve.

Returværdi: Ingen

Se også: COLORF

Eksempel:

```
COLOR("#3-4",255,0,0) /* Felt 3 og 4 får en rød box  
COLOR("#3-4",-1)    /* Ingen baggrund for felterne
```

6.5. **COLORF** - Sæt forgrunds tekst farve for et antal felter

COLORF(felter *par1*, FarveRød *par2*, FarveGrøn *par3*, FarveBlå *par4*)

par4 : Blå farveværdi (0-255)

Beskrivelse: Forgrundsfarven for de angivne felter sættes til RGB værdien, dvs. teksten i feltet bliver den angivne farve.

Returværdi: Ingen

Se også: COLOR

Eksempel: COLORF("#3-4",0,0,255) /* Felt 3 og 4 udskrives med blå tekst

6.6. DIALOG - Ekstra input dialog

Tal DIALOG(Felter *par1*)

Parametre: *Par1*: Felter der skal vises i dialogen

Beskrivelse: DIALOG funktionen sætter brugeren i stand til at vise dialogboks med udvalgte felter på ethvert tidspunkt i en rapport eller i et IQ program, her for eksempel ved klik på et felt.

DIALOG("#1,7-8,le#3") definerer en dialog med de angivne felter. Feltenes dokumentation anvendes som flydende online hjælpetekster, der fremkommer, når musemarkøren flyttes hen over feltnavnene.

Sammen med feltnumrene kan angives en eller flere af følgende optioner:

Lxxxx	Linie	(dialog enheder)
Pxxxx	Position	(dialog enheder)
Hxxxx	Højde	(dialog enheder)
Wxxxx	Bredde	(dialog enheder)
N	Ingen overskrift	
N1	Tilføj feltnummer til overskriften	
N2	Vis overskriften over feltet istedet for til venstre for dette	
C	COMBOBOX, Felt check definitioner vises som værdier heri	
O	LISTBOX, Felt check definitioner vises som værdier heri	
:xx	Skip til næste kolonne feltlinie xx	
+xx	Skip xx feltlinier ned	

Returværdi: OK=0, FORTRYD=1

Se også: PARAMS

Eksempel:

```
DIALOG("#1-3,11")          /* Opbyg en dialog med de angivne felter
```

6.7. GETFLAG- Aflæs parametre for et felt (IQ)

tal CLRFLAG(felter *par1*, tal *par2*, tal *par3*)

Beskrivelse: Hvert felt på skærmen er tilknyttet et antal parametre (bits), der definerer, hvordan feltet anvendes. SETFLAG funktionen kan anvendes til at sætte disse parametre, CLRFLAG til at deaktivere dem.

Returværdi: Ingen

Se også: SETFLAG, CLRFLAG

Eksempel: GETFLAG("#12,44",7,0)

6.8. SETFLAG- Sæt parametre for skærmfelter (IQ)

SETFLAG(*felter par1*, *Bitflag par2*, *tal par3*)

Beskrivelse: Hvert felt på skærmen er tilknyttet et antal parametre (bits), der definerer, hvordan feltet anvendes. SETFLAG funktionen kan anvendes til at sætte disse parametre, CLRFLAG til at deaktivere dem.

For typeparameter 3 skal normalt altid angives 0.

Returværdi: Ingen

Se også: GETFLAG, CLRFLAG

Eksempel: SETFLAG("#12,44",7,0)

6.9. **ZERO** - Nulstil et antal felter

ZERO(felter *par1*)

Parametre: *par1* : Feltangivelse

Beskrivelse: De angivne felter nulstilles. ZERO fungerer på samme måde som LET.

Returværdi: Ingen

Se også: LET, CLEAR

Eksempel: ZERO("3,19") /* Nulstiller felt 3 og felt 19

7. Rapport kontrol

Dette kapitel beskriver funktioner, der kan anvendes til styring af flowet i beregninger/udskrift for en rapport i RAPGEN. Funktionerne CHAIN, MESS og RETURN kan ligeledes anvendes i IQ og DATAMASTER, de øvrige funktioner er ikke relevante for skærmprogrammer.

7.1. **CHAIN** - Start af næste rapport eller andet program (RAP)

tal CHAIN()

par3 : Blank eller Index, Totalniveau, Firmanr

Beskrivelse: CHAIN(7) vil starte rapport 7 efter denne rapport er færdig. Ved starten anvendes samme opstartsparemetre som for denne rapport.

CHAIN(7,"310395,-,9999","1") sætter Pr.dato til 310395, Startfra til ingenting, Stopved til 9999 samt laveste totalniveau til 1. De øvrige startparametre er uændret.

CHAIN(2007) starter rapport nummer 7 i subsystem 2.

CHAIN(-1,"c:/windows/write.exe") starter dette (windows)program.

Hver gang CHAIN(nr) udføres tildeles et nyt kørselsnummer fra 1 og fremefter, hvis en rapport er startet fra menuen er kørselsnummeret 0. Med #20=CHAIN() hvor CHAIN anvendes uden parametre, kan man hente dette nummer og således bringe en rapport til at køre et antal gange, for eksempel udskrive et antal kopier.

CHAIN("c:/windows/write.exe") kan anvendes i IQ/DATAMASTER programmer til at starte et andet windowsprogram.

Returværdi: CHAIN() returnerer det aktuelle kørselsnummer.

Se også: EXIT, CHAINR

Eksempel:

```
#20=CHAIN()          /* Dette er rapportnummer 7  
IF #20<3 CHAIN(7)    /* Den samme rapport startes 4 gange efter hinanden.
```

7.1.1. CHAINR- Start program eller extern kommando direkte (RAP)

par3 : Blank eller Index, Totalniveau, Firmanr

Beskrivelse: CHAIN kommandoen vil altid blive placeret SIDST i rapporten, dvs. at det næste program først startes efter det kørende er afsluttet.

Brug istedet CHAINR for at afbryde det kørende program og aktivere et andet her og nu.

Returværdi: Ingen

Se også: EXIT , CHAIN

Eksempel: CHAINR(-1,"Notepad") /* Start notepad lige nu

7.1.2. **CHAIN** - Start IQ program eller extern kommando (IQ)

CHAIN(text *par1*, text *par2*)

par2 : Eventuel nøgle for læsning af record

Beskrivelse: Aktiver et programnummer eller en Windows kommandostreng.

Returværdi: Ingen

Se også: EXIT, ISACTIVE, WAIT

Eksempel:

```
CHAIN ("20")      starter program 20.  
CHAIN ("+5")      starter program 5 og aktiverer dette.  
CHAIN (">5")      starter program 5, den viste record vil ikke blive overført  
CHAIN ("$5")      starter program 5, aktiverer det og venter indtil færdigt  
CHAIN ("+5",#1)   starter program 5 som vil læse en record med nøgle #1
```

```
#20="notepad"
```

```
#20="command.com /C edit minfil.txt"
```

```
CHAIN(#20)        starter det angivne Windows program
```

```
CHAIN("rapwin &") & som sidste karakter lader IQ fortsætte  
                  samtidig med de nystartede program kører.
```

7.2. **WAIT** - Vent til et program er færdigt (IQ)

WAIT(programnr *par1*)

Parametre: *par1* : Programnummer

Beskrivelse: Vent indtil det givne program er færdigt (se EXIT). Beregningerne fortsætter først, når det ander programvindue lukkes.

Returværdi: Ingen.

Se også: CHAIN , EXIT

Eksempel: WAIT(20) */* Fortsæt ikke før program 20 afsluttes*

7.3. **COMPILE** - Kompiler rapport (RAP)

COMPILE(tal *par1*)

Denne funktion kan kun anvendes, såfremt man har en C compiler installeret og RAPGEN er købt med licens for kompilering.

Beskrivelse: I stedet for at skulle vælge 'Compile' fra 'Parameter' menuen, hver gang rapporten startes efter ændringer, kan dette fastlægges med en beregning.

Se også: INSTALL

Eksempel: COMPILE /* Rapporten kompileres

7.4. **EXIT** - Afslut rapport (RAP)

tal EXIT(tal *par1*)

Beskrivelse: Funktionen afslutter rapporten eller det aktuelle gennemløb (sortering/print).

Se også: CHAIN, CHAINR, MESS

Eksempel:

```
READ(le)           /* Læs leverandørens data
IF #OK THEN BEGIN  /* afslut rapporten hvis leverandøren ikke findes
#12="Leverandøren ", le#1, " findes ikke:"
MESS(#12)
EXIT(2)
END
```


7.4.1. **EXIT** - Luk et IQ program eller vindue (IQ)

EXIT(tal *par1*)

Parametre: *par1* : Programnummer, der skal lukkes

Beskrivelse: EXIT(0) lukker det aktive IQ program.

Returværdi: Ingen

Se også: CHAIN , MESS, WAIT

Eksempel:

EXIT(20) lukker program 20 hvis dette er åbent, 1020 giver subsystem 1.

EXIT(-1) lukker programvalgsvinduet.

EXIT(-2) lukker feltvalgsvinduet for definition af et nyt program.

EXIT(-3) lukker og afslutter hele IQ.

7.5. KEYS - Start/Stop angivelser (RAP)

tal KEYS()

par2 : Eventuelt fast navn på .KEY definitionsfil

Beskrivelse: Med KEYS funktionen kan man bringe en rapport til at køre over en række start/stop intervaller defineret som linier i en tekstfil. KEYS erstatter herved angivelsen af START/STOP ved opstart samt eventuelt også INDEX.

Keysfilen kan oprettes med en simpel tekst-editor og kan for eksempel indeholde:

```
0001
1000-1999
0005-0099,0200,0155-0157
2:205-271
47/2000-2500
```

Der kan for hver linie angives enkelt nøgler eller intervaller til udskrift. Med 2: angives kørsel via index 2, med 47/ angives en beregningskode, der kan hentes med for eksempel #20=KEYS() og herefter anvendes i beregningerne.

Anvendes KEYS(0) fås en samlet liste med alle de angivne intervaller, KEYS(1) bevirker at der udskrives en selvstændig rapport med total for hver linie i keysfilen, ENDSUM funktionen kan eventuelt anvendes for at få en grande total.

Det er ikke absolut nødvendigt at anvende KEYS i en rapport, der skal styres på denne måde. Ved opstart af rapporten, i START FRA, kan man angive:

```
(aa)                               Start med keysfil aa
(1000,1100-1200,0004)              Kør over disse intervaller
```

Angives ingen sti/extension for keyfilen vil denne være placeres på det normale rapportdirektory med extension .KEY, for eksempel c:/rapfil/rap/aa.key

Returværdi: KEYS() returnerer beregningskoden (47 ud af 47/111-222) for det nuværende interval.

Se også: ENDSUM, INDEX

Eksempel:

```
KEYS(0,"c:/mydir/enfil.min")      /* Rapporten styres via denne fil.
#20=KEYS()                          /* En styrekode hentes
```

7.6. **INDEX** - Sæt index og start/stop værdi for rapport (RAP)

tal INDEX(index *par1*, tekst *par2*, tekst *par3*)

par3 : angiver den værdi brugeren normalt indtaster i feltet Stop ved

Beskrivelse: Funktionen benyttes til fast at sætte indeks og start/stop interval for en rapport. Såfremt *par1* ≥ 1 sættes indekset for rapportens hovedkartotek, dvs. den orden kartotekets poster skal læses i. Hvis *par2* indeholder noget vil funktionen sætte start-intervallet og tilsvarende for *par3*.

Hvis start fra/stop ved første karakterer er plus (+) sættes den angivne værdi foran det, brugeren indtaster ved start af rapporten.

INDEX(-2) låser rapporten til index 2 i faldende sorteringsorden. Database driveren skal supportere læsning i faldende orden.

Returværdi: Returnerer det indeks som hovedkartoteket benytter.

Se også: KEYS

Eksempel: INDEX(2,"D","D") /* rapportens hovedkartotek er KU (demo-kurser)

Eksemplet fastfryser rapporten til indeks 2, således at kurserne læses sorteret efter valutanavn og ikke valutakode. Ydermere indlæses kun de poster, hvor navnet starter med "D".

INDEX(1,"+02","+02") /* Udskriv 024711 når der tasteres 4711

7.7. **LTOT** - Sæt laveste total niveau (RAP)

tal LTOT(Niveau *par1*)

Parametre: *par1* : angiver det laveste totalniveau som ønskes for rapporten

Beskrivelse: Såfremt *par1* ≥ 0 vil funktionen sætte laveste totalniveau for rapporten. Dette niveau svarer til det totalniveau der indtastes ved start af en rapport.

Returværdi: Returnerer rapportens laveste totalniveau.

Se også: MTOT

Eksempel: LTOT(1) /* *udskriv kun totaler, undertryk alle enkeltlinier*

7.8. **MTOT** - Sæt det maksimale totalniveau (RAP)

tal MTOT(Niveau *par1*)

Parametre: *par1* : angiver det maksimale totalniveau for rapporten

Beskrivelse: Funktionen fastlåser det maksimale totalniveau for rapporten. Såfremt *par1* er lig 0, vil rapporten ikke udskrive nogen totaler.

Returværdi: Returnerer det maksimale totalniveau.

Se også: LTOT

Eksempel: MTOT(1) */* En nonsens grande total undertrykkes*

7.9. **MESS** - Skriv meddelelse på skærmen

tal MESS(*tekst par1*)

Parametre: *par1* : angiver den meddelelse, som skal skrives på skærmen

Beskrivelse: MESS viser en meddelelsesbox på skærmen. Afhængig af den sidste karakter i teksten vises følgende symbol og knapper:

Tekst	Symbol	Knapper	Defaultknap
tekst	Info	OK	OK
tekst?	!	OK, CANCEL	CANCEL
tekst??	?	YES, NO, CANCEL	YES
tekst!	!	YES, NO	YES
tekst!!	STOP	OK	OK
tekst?!	STOP	OK, CAN	OK

Returværdi: 0=OK eller YES, 1=NO, -1 = CANCEL

Se også: EXIT

Eksempel:

```
#1=MESS("Skal rapporten stoppes !")
IF #1=0 EXIT(0) /* afslut rapporten
```

7.10. NOPAS - Intet password/brugernavn på rapporten (RAP)

NOPAS()

Parametre: Ingen

Beskrivelse: Funktionen fjerner password/kendeord beskyttelse fra rapporten. Normalt vil en rapport, der foretager opdatering få password CARE. Ved brug af NOPAS() eller PAS() kan dette password enten fjernes eller sættes til et andet.

Se også: PAS, UPDATE

Eksempel:

```
UPDATE (1)
```

```
NOPAS ()
```

```
/* intet password for denne rapport
```

7.11. **PAS** - Sæt password/brugernavn (RAP)

tal PAS(tekst *par1*)

Parametre: *par1* : angiver det ønskede password/kendeord

Beskrivelse: Funktionen sætter et fast password/kendeord for rapporten. Dette skal brugeren indtaste med start af rapporten.

Se også: NOPAS

Eksempel: PAS("SWTOOLS") /* sæt password til SWTOOLS

7.12. PARAMS - Ekstra rapport start parametre (RAP)

PARAMS(Felter *par1*)

Parametre: *Par1*: Felter der skal vises i startdialogen

Beskrivelse: PARAMS("#1,7C,60,le#3") er en variant af dialog funktionen, hvor input foretages ved start af en rapport, ikke under selve kørselen af denne.

Brug af PARAMS i beregningerne på en rapport vil indsætte en ny knap <Ekstra parameter> på startskærmbilledet for rapporten, ved tryk på denne vil dialogen med felterne blive vist.

Returværdi: Ingen.

Se også: DIALOG

Eksempel:

```
PARAMS("#1-3,11")
```

```
/* Opbyg en dialog med de angivne felter
```

7.13. **RETURN** - Returner fra beregninger

tal RETURN(tal *par1*)

Parametre: *par1* : angiver den kode som skal returneres

Beskrivelse: Funktionen benyttes til at afslutte beregningerne for den aktuelle post læst fra hovedkartoteket. Hvis ingen parameter angives eller *par1* lig 0, vil rapporten udskrive de definerede rapportlinier for posten. Returneres f.eks 1 vil den aktuelle post ikke blive behandlet og derfor ikke udskrevet.

Returværdi: Ingen.

Se også: GOSUB

Eksempel: IF LE#6 < 1000 RETURN(1) /* ingen udskrift hvis saldo < 1000

7.14. **SORTKEY** - Frigivelse af extra sorteringsnøgle (RAP)

tal SORTKEY(kartotek *par1*)

Parametre: *par1* : 0, -1 eller kartoteksid

Beskrivelse: Under visse omstændigheder kan man ønske at sortere en liste, således at hver record optræder flere gange, for eksempel en vareliste, hvor varen skal findes både under den normale leverandør og under en eventuel alternativ leverandør.

I dette tilfælde sorterer man efter et frifelt, der herefter beregnes og en extra sorteringsnøgle indsættes hver gang SORTKEY funktionen kaldes.

Flere kartoteker kan endvidere blandes med denne funktion. Sorteringskartoteket indeholder et nummer, der normalt peger på en record i rapportens hovedkartotek. Med SORTKEY(*le*) indsættes en nøgle, der peger på kartotek *le*, med #20=SORTKEY(-1) kan man aflæse, hvilket kartotek, der i øjeblikket fungerer som hovedkartotek, og styre beregningerne efter dette.

Returværdi: Hovedkartoteknummer, normalt 1.

Se også: MERGE

Eksempel:

```
#11=#9          /* Sorteringsfrifelt = Alternativ leverandør
IF #11<>0 SORTKEY(0) /* Extra sorteringsnøgle med denne
#11=#6          /* Normal sortering iflg.lleverandør
```

7.15. SORTWORK - Anvendelse af bestemt sorteringskartotek (RAP)

SORTWORK(tal *par1*)

Parametre: *par1* : Sorteringskartotek nummer

Beskrivelse: Ved sortering opretter RAPGEN hjælpekartoteker med navnene: c:/tmp/SIN00000.000 og c:/tmp/SUD00000.000 hvor c:/tmp/ er den normale TMP sti. Disse sorteringskartoteker slettes ikke efter kørsel idet man ved næste rapportstart ved angivelse af

START VED: SORT eller SORTD

kan undgå sorteringstiden og istedet bruge samme sortering som sidst. Anvendes dette hyppigt kan man sikre sig mod sletning af hjælpekartotekerne ved at angive et nummer med for eksempel SORTWORK(47) hvorved navnene bliver: c:/tmp/SIN00000.047 og c:/tmp/SUD00000.047

Returværdi: Ingen.

Se også:

Eksempel: SORTWORK(47)

7.16. **NÅR** - Hvornår der skal beregnes (RAP)

NÅR(*tal par1*, *tal par2*)

Beskrivelse: Kommandoen NÅR kan anvendes, når der er flere muligheder for at få udført beregningerne.

8. Printer styring

I dette afsnit beskrives de forskellige printerfunktioner.

8.1. **COPIES**- Antal kopier af udskriften (RAP)

COPIES(tal *par1*, Printer *par2*)

par2 : Eventuelt printer nummer

Beskrivelse: COPIES(1) giver een ekstra kopi af udskriften. Op til 30 kopier kan angives og der skal være plads til alle disse Windows printfiler

COPIES(1,7) producere een ekstra kopi på printeren defineret som nummer 7 i printer opsætningen. Bemærk dog at uventede sideskift vil forekomme hvis kopiprinteren har en mindre sidehøjde end hovedprinteren

Returværdi: Ingen.

Se også: PRINTER

Eksempel: COPIES(1) /* *Print 2 times*

8.2. PAGE - Skift layout side (RAP)

tal PAGE(tal *par1*)

Parametre: *par1* : angiver den ønskede rapport side

Beskrivelse: Normalt når en rapport udskriver benytter den sig af rapport side 0, svarende til det layout defineret i RAPGEN. Men hvis rapporten f.eks skal udskrive på et andet sprog kan rapporten rumme op til 9 forskellige layouts. Disse layouts ligger placeret på side 0-9 og kan nås fra 'fil' menuen, 'side layout' i ændring af layout.

Returværdi: Returnerer den side som benyttes som layout.

Se også: PRINT

Eksempel:

```
PAGE(1e#5)      /* skift layout side udfra leverandørens sprog  
PRINT(1-10)    /* skriv tekst
```


8.3. **PRINT** - Udskrift af linier fra rapportens layout (RAP)

tal PRINT(*tekst par1*)

Parametre: *par1* : angiver de linier som skal udskrives

Beskrivelse: Funktionen benyttes til udskrivning af linier fra rapportens layout, eller til at sætte printkommandoer, der bl.a udføres ved sideskift og udskrift af totallinier. Den syntaks der skal benyttes ved udskrift af linier er:

Funktion	Beskrivelse
PRINT(1-10)	Udskriv linierne 1 til 10
PRINT(1,+2,2)	Udskriv linie 1, herefter 2 blanke linier og til sidst linie 2
PRINT(1,:60,2)	Udskriv linie 1, gå til linie 60 på printeren og udskriv linie 2
PRINT(:1003,1,3)	Gå til 3 linier før bunden og udskriv linie 1 og 3
PRINT(1-10,:1,20)	Udskriv linie 1 til 10, skift side og udskriv linie 20
PRINT(*H)	Linierne defineret med H= udskrives

Funktionen benyttes ligeledes til af sætte print-kommandoer, der bl.a fortæller hvilke linier rapporten skal udskrive ved sideskift:

Funktion	Beskrivelse
PRINT(H=1-4)	Ved sideskift skal linier 1 til 4 udskrives
PRINT(L=8)	Den normale printlinie som udskrives for hver post er printlinie 8
PRINT(T=10)	Totallinien som bl.a. udskrives ved grand totalen er linie 10
PRINT(D=9)	Overskrift for Detaillinier(READH) bliver linie 9
PRINT(B=:1002,17)	Som Bund på hver side udskrives linie 17
PRINT(N=3,:1,1-4)	Ny side 3 linier fra sideslut, overskrift linie 1-4
PRINT(A=10)	Linie 10 udskrives før en totalblok
PRINT(C=11)	Linie 11 udskrives efter en totalblok

Bemærk at et tekstfelt kan anvendes som parameter for Printkommandoen som fx:

```
#11="1-4, 15"  
PRINT(#11)
```

PRINT(>2) skifter til printer 2, se PRINTER.

Returværdi: Ingen.

Se også: PAGE, PRINTER;

Eksempel: PRINT(:60,1-10) /* gå til linie 60 og udskriv linie 1 til 10

8.3.1. **PRINT** - Printer udskriffs kontrol (RAP.)

PRINT(tekst *par1*)

Parametre: *par1* : Option=værdi

Beskrivelse: PRINT kommandoen er udvidet med syntaksen PRINT(xx=værdi yy), hvor xx,værdi og yy kan være een af følgende:

	Funktion	Beskrivelse
xx=	ml	Venstre margin
	mr	Højre margin
	mt	Top margin
	mb	Bund margin
	eh	Højden af en tom linie
	ce	Luk rapportvinduet ved afslutning
	fh	Standard skrifthøjde for alle linier
	cd	Luk printeren og start på en ny udskrift
yy=	cm	Centimeter
	in	Tommer
	pt	Punkter
<ingen>	Device pixels	

8.3.2. PRINT(=? - Aflæs printer opsætning (RAP.)

PRINT(=?tekst *par1*)

Beskrivelse: PRINT kommandoen er også udvidet med en funktion til at kunne aflæse visse oplysninger om printerens.

Den returnerede værdi i yy gives som punkter undtagen for xx= 5, 8, 9, 15 eller 16.

8.4. PRINT(LAB= - Label funktion (RAP)

PRINT(LAB=Tekst *par1*, Tekst *par2*, Tekst *par3*, Tekst *par4*, Tekst *par5*, Tekst *par6*)

par6 : Antal

Beskrivelse: Højden og bredden for en label på arket kan angives i centimeter eller tommer med:

7cm	svarende til 7 centimeter
2in	svarende til 7 tommer

I eksemplet nedenfor udskrives labels fra venstre mod højre på et label ark med 21 labels, 3 på hver række og 7 rækker, hvor hver label måler 7*7 cm. Der udskrives 2 stk. af hver label.

Returværdi: Ingen.

Se også: PRINT

Eksempel:

```
FØRST
PRINT(LAB=1,3,7,7cm,7cm,2) /* Definer label udskrift
NORMAL
```

8.5. PRINTER- Printervalg (RAP.)

PRINTER(Printer *par1*)

Parametre: *par1* : Printernummer

Beskrivelse: Denne funktion bruges i sammenhæng med printervalget ved start. For at sætte standardprinterens for en rapport kan nedenstående beregningslinie indføres:

Returværdi: Ingen

Se også: COPIES, PRINT

Eksempel: PRINTER(7) /* Standard printeren for denne rapport er printer 7

8.5.1. PRINTER - Udskrift på flere printere (RAP)

PRINTER(tal *par1*, Printer *par2*)

Parametre: *par1* : Printernummer *par2* : PrinterID

Beskrivelse: PRINTER(2,7) åbner anden printer som den printer, der er defineret som printernummer 7 i printeropsætningen. Der skrives intet på denne printer før

PRINT(>2)

mødes som en beregningslinie hvorefter al udskrift foretages herpå indtil PRINT(>1) skifter tilbage til den printer, der er valgt ved start af rapporten.

Hver printer has sine egne sidenumre og kan have forskellig papirstørrelse. Op til 30 printere (eller kopier) kan anvendes samtidig.

Returværdi: Ingen

Se også: COPIES, PRINT

Eksempel: PRINTER(2,7) /* Åben anden printer nummer 7

8.6. **PRTTOTAL** - Manuel styring af totaludskrift (RAP)

PRTTOTAL(Niveau *par1*)

Parametre: *par1* : Total niveau nummer

Beskrivelse: Rapportgeneratoren udskriver normalt en subtotal når en del af sorteringskriteriet for rapporten skifter. Med PRTTOTAL kan man frakoble denne funktion og istedet angive at subtotaler skal udskrives når et bestemt felt skifter.

Returværdi: Ingen.

Se også: ENDSUM

Eksempel:

```
IF #7=1 PRTTOTAL(1)          /* Udskriv subtotal hvis felt 7 er 1
SIDST
PRTTOTAL(2)                 /* Udskriv grandetotal til sidst
```

8.7. SCRVRT - Genkald gemt skærmprint (IQ)

SCRVRT(Filnavn *par1*)

Parametre: *Par1*: Filnavn der indeholder det gemte skærmprint

Beskrivelse: SCRVRT("filnavn") kalder skærmprinterens og viser det print, der er gemt i den angivne fil. Dette kan for eksempel indbygges i et IQ program ved klik på et felt.

Returværdi: Ingen.

Se også: PRINT

Eksempel:

```
SCRVRT("c:/w/ab.cde")      /* Vis indholdet af denne fil med skærmprinterens
```


9. Læsning af kartoteker

Dette kapitel beskriver READ funktionen for læsning af et sekundært kartotek samt START/NEXT/OM funktionerne for læsning af flere records.

Grundprincipperne for kartoteksforbindelser er beskrevet i RAPGEN brugerhåndbog, afsnittet om flere kartoteker.

9.1. READ - Læs en post fra et kartotek

tal READ(kartotek *par1*, index *par2*) ,forbindelse *par3*

par3 : angiver en eventuel forbindelse, der ikke findes som standard

Beskrivelse: Funktionen læser en post fra et kartotek.

READ(le) vil læse kartotek le ved hjælp af den standardforbindelse, der er defineret i Data Dictionaryet.

READ(le),#9 vil læse kartotek le med #9 som nøgle til index 1 uanset om der findes nogen standardforbindelse angivet.

READ(va.02),#6 vil læse kartotek va med felt 6 som nøgle til index 2, uanset om der findes nogen standardforbindelse angivet.

READ(le),"1",#9(3,4),#7 vil danne en nøgle som en kombination af konstanten "1", felt 9 karakter 3-4 og felt 7.

READ(le.00),#6 vil læse le med recordnummeret (index 0) som givet i felt 6.

Returværdi: 0 hvis posten er læst.

Se også: START, NEXT, OM, END, PRIOR, READR, READX

Eksempel: READ(le) /* læs leverandøren

9.2. **READH** - Læs en post og udskriv eventuelt overskrift

tal READH(kartotek *par1*, index *par2*) ,forbindelse *par3*

par3 : angiver en eventuel forbindelse, der ikke findes som standard

Beskrivelse: Funktionen læser en post fra et kartotek, præcis som READ.

Såfremt der læses en anden post end sidste gang, READH blev anvendt, for eksempel når leverandørnummeret skifter, udskrives de overskriftslinier, der er angivet i for READH.

Returværdi: 0 hvis posten er læst.

Se også: READ

Eksempel: READH(1e) */* læs leverandøren med eventuel overskrift*

9.3. **READR** - Læs record med bestemt recordnummer

tal READR(kartotek *par1*) ,forbindelse *par2*

par2 : angiver en eventuel forbindelse, der ikke findes som standard

Beskrivelse: Funktionen læser en post fra et kartotek med angivelse af recordnummer. READR kan kun anvendes på databaser, der arbejder med recordnumre, og er kun medtaget af hensyn til kompatibilitet med tidligere versioner.

READ(le.00),#6 er det samme som READR(le),#6

Se også: READ, READX

9.4. **READX** - Læs record med relativ recordnummer angivelse

tal READX(kartotek *par1*) ,forbindelse *par2*

par2 : angiver en eventuel forbindelse, der ikke findes som standard

Beskrivelse: Funktionen læser en post fra et kartotek med angivelse af relativt recordnummer. READX kan kun anvendes på databaser, der arbejder med recordnumre, og er kun medtaget af hensyn til kompatibilitet med tidligere versioner.

READ(le.00),#6+N er det samme som READX(le),#6

Se også: READ, READR

9.5. **START** - Sæt index og interval for et kartotek

tal `START(kartotek par1, index par2) ,forbindelse par3`

par3 : angiver en eventuel forbindelse, der ikke findes som standard

Beskrivelse: Funktionen forbereder læsning med `NEXT` funktionen ved at definere det interval af nøgler, denne skal anvende.

Standardforbindelsen mellem kartotekerne kan anvendes, eller man kan angive en individuel nøgle som beskrevet under `READ`.

Ved `START` udfyldes nøglen til kartoteket normalt ikke helt. Efterfølgende læsninger med `NEXT` vil finde alle records, hvor den første del af nøglen passer med den i `START` angivne del.

Returværdi: 0 hvis interval sat.

Se også: `READ`, `NEXT`, `OM`, `END` , `PRIOR`

Eksempel:

```
#47=0                /* Nulstil sumfelt
START(va)            /* Start læsning af varer
NEXT(va)             /* Læs den næste vare
#47=#47+va#3        /* Summer alle varer
OM(va)               /* Fortsæt indtil ikke flere varer
```

9.6. NEXT - Hent næste post i interval

tal NEXT(kartotek *par1*)

Parametre: *par1* : angiver forkortelsen på kartoteket

Beskrivelse: Funktionen benyttes i sammenhæng med START/NEXT/OM løkker. Ved funktionerne START() og END() sættes det ønskede interval for løkken. NEXT() læser herefter een post fra kartoteket. Når beregningslinien OM() udføres vil funktionen NEXT() igen udføres indtil der ikke findes flere poster i intervallet.

Returværdi: Returnerer 0 så længe der er poster i intervallet.

Se også: READ, START, OM, END , PRIOR

Eksempel:

```
PRINT                /* Overtag al udskriftsstyring selv
PRINT(4,6,5)         /* Udskriv leverandør-informationer
START(va)            /* Start læsning af varer
NEXT(va)             /* Læs den næste vare
PRINT(7)             /* Udskriv varelinierne
OM(va)               /* Fortsæt indtil ikke flere varer
```

9.7. OM - Gentag læsning med NEXT

tal OM(kartotek *par1*)

Parametre: *par1* : angiver forkortelsen på kartoteket

Beskrivelse: Funktionen benyttes i sammenhæng med START/NEXT/OM løkker. Ved funktionerne START() og END() sættes det ønskede interval for løkken. NEXT() læser herefter een post fra kartoteket. Når beregningslinien OM() udføres vil funktionen NEXT() igen udføres indtil der ikke findes flere poster i intervallet.

Returværdi: Ingen.

Se også: START, NEXT PRIOR

Eksempel:

```
#47=0          /* Nulstil sumfelt
START(va)      /* Start læsning af varer
NEXT(va)       /* Læs den næste vare
#47=#47+va#3   /* Summer alle varer
OM(va)         /* Fortsæt indtil ikke flere varer
```


9.8. **GETKEY** - Hent den nuværende nøgle

tekst GETKEY(kartotek *par1*)

Parametre: *par1* : Kartoteksforkortelsen

Beskrivelse: #20=GETKEY(va) henter indexnøglen for den sidst læste record i kartoteket va. Funktionen anvendes specielt for database systemer, hvor nøglen ikke nødvendigvis behøver at findes som et felt i databasen.

Returværdi: Nøgleværdien på tekstform.

Se også:

Eksempel: #20 = GETKEY(va)

9.9. **END** - Sæt slut interval efter START

tal END(kartotek *par1*) ,forbindelse *par2*

par2 : angiver højeste nøgle i intervallet.

Beskrivelse: START sætter første og sidste nøgle i et interval ens, således at for eksempel alle poster med samme debitornummer læses.

Normalt behøver man ikke anvende END, kun hvis der ønskes en anden slut på intervallet end START har sat.

Returværdi: Ingen.

Se også: READ, START, OM, NEXT , PRIOR

Eksempel:

```
UPDATE (1)                /* Nulstilling af arbejdskartotek xx
START (xx) ,"0000"        /* Start læsning af kartoteket helt forfra
END (xx) ,"9999"         /* Gå helt til slut
NEXT (xx)                 /* Læs den næste record
DELETE (xx)               /* Slet alle
OM (xx)                   /* Fortsæt indtil kartoteket er tomt
```

9.10. **PRIOR** - Hent forrige post i interval

tal PRIOR(kartotek *par1*)

Parametre: *par1* : angiver forkortelsen på kartoteket

Beskrivelse: PRIOR fungerer præcis som NEXT, blot findes ikke næste men forrige record.

NB: Ikke alle databasesystemer supporter læsning i 'omvendt' orden.

Returværdi: Returnerer 0 så længe der er poster i intervallet.

Se også: READ, START, OM, NEXT, END

Eksempel:

```
PRINT                               /* Overtag al udskriftsstyring selv
#47=0                               /* Nulstil tæller
START(va)                           /* Start læsning af varer
PRIOR(va)                           /* Læs forrige vare
#47=#47+1                           /* Tæl antal varer
IF #47=1 PRINT(4,6,5)              /* Udskriv leverandør-hoved første gang
PRINT(7)                            /* Udskriv varelinierne i omvendt orden
OM(va)                              /* Fortsæt indtil ikke flere varer
IF #47>0 PRINT(8)                 /* Udskriv bund hvis der var varer
```

9.11. **SPEED**- Optimering af læsestrategi

SPEED()

Parametre: Ingen

Beskrivelse: SPEED() funktionen kan anvendes til at optimere læsestrategi på en rapport idet en record med samme nøgle som sidst anvendt ikke vil blive læst igen, men taget fra den interne buffer fra sidste læsning. Man skal dog være yderst forsigtig med dette, såfremt der er tale om en rapport med opdatering af kartoteker.

Returværdi: Ingen

Se også: READ

Eksempel: SPEED() */* Optimering af READ på en rapport*

10. Skrivning i kartoteker

Dette kapitel beskriver de forskellige muligheder for skrivning i filerne. Det er en forudsætning for at kunne anvende disse funktioner, at der ved installation er fortaget med skrive tilladelse, at den benyttede database supporterer skrivning, samt at brugeren på serveren har tilladelse hertil.

Et program, der opdaterer kartoteker, bør altid testes før brug. Det er **brugerens eget ansvar** at opdateringen er testet og foretages korrekt.

10.1. UPDATE - Tilladelse til skrivning i kartoteker

tal UPDATE(tal *par1*, felter *par2*)

par2 : Tilladte felter.

Beskrivelse: UPDATE(1) skal indsættes i en opdaterende rapport, før nogen af skrivefunktionerne anvendes, for at aktivere disse.

UPDATE kommandoen er udvidet med specifikation af, hvilke felter, der må opdateres.

```
UPDATE (1,"va#6") /* Tillader kun opdatering af felt 6 i kartoteket va
```

```
UPDATE (1,"le#3-4") /* Hvert kartotek skal specificeres på sin egen linie
```

```
UPDATE (0) /* Kan nu bruges i DATAMASTER til at frakople al skrivning
```

Returværdi: Ingen.

Se også: DELETE, INSERT, REWRITE, WRITE, NOPAS

Eksempel:

```
UPDATE (1) /* rapporten opdaterer
```

```
NOPAS () /* intet password
```

```
#6=#6+10 /* Beregning af nye feltværdier
```

```
REWRITE (1e) /* opdater leverandøren i kartoteket
```

10.2. REWRITE - Opdater en eksisterende post i kartoteket

tal REWRITE(kartotek *par1*)

Parametre: *par1* : angiver forkortelsen på kartoteket

Beskrivelse: Funktionen opdaterer en post i det angivne kartotek, der skal være indlæst først. Kun hvis databasesystemet supporterer dette, kan værdien indexfelter ændres. For at være aktiv skal beregningslinien UPDATE(1) være udført.

Returværdi: 0 hvis posten er blevet opdateret.

Se også: DELETE, INSERT, WRITE, NOPAS, UPDATE

Eksempel:

```
UPDATE (1)          /* rapporten opdaterer
NOPAS ()            /* intet password
EFTER              /* KUN EFTER SELEKTION
#6=#6+10           /* Beregning af nye feltværdier
REWRITE(1e)        /* opdater leverandøren i kartoteket
```

10.3. INSERT - Indsæt ny post i kartotek

tal INSERT(kartotek *par1*)

Parametre: *par1* : angiver forkortelsen på kartoteket

Beskrivelse: Funktionen indsætter en ny post i et kartotek. ALLE felter i kartoteket skal være sat til en værdi før INSERT foretages. For at være aktiv skal beregningslinien UPDATE(1) være udført.

Returværdi: 0 hvis posten er indsat.

Se også: DELETE, REWRITE, WRITE, NOPAS, UPDATE , CLEAR, LET

Eksempel:

```
UPDATE (1)           /* der opdateres
NOPAS ()             /* intet password
CLEAR (1e)          /* nulstil alle felter for leverandøren
LET ("1e#1,3=#7,17") /* udfyld værdierne
INSERT (1e)         /* indsæt ny leverandør i leverandørkartotek
```


10.4. DELETE - Slet en post i et kartotek

tal DELETE(kartotek *par1*)

Parametre: *par1* : angiver forkortelsen på kartoteket

Beskrivelse: Funktionen sletter fysisk en post i det ønskede kartotek. Denne skal være læst, før sletning kan foretages. For at være aktiv skal beregningslinien UPDATE(1) være udført.

Returværdi: 0 hvis posten er slettet.

Se også: INSERT, REWRITE, WRITE, NOPAS, UPDATE

Eksempel:

UPDATE (1)	<i>/* der opdateres</i>
NOPAS ()	<i>/* intet password</i>
EFTER	<i>/* KUN EFTER SELEKTION</i>
DELETE (va)	<i>/* De udvalgte varer slettes</i>

10.5. WRITE - Opdater eller indsæt post i kartotek

tal WRITE(kartotek *par1*)

Parametre: *par1* : angiver forkortelsen på kartoteket

Beskrivelse: Funktionen opdaterer eller indsætter en post i det angivne kartotek. Såfremt sidste READ på kartoteket fandt en record, vil funktionen fungere som REWRITE(), fandtes ingen record ved READ fungerer denne som INSERT(). For at være aktiv skal beregningslinien UPDATE(1) være udført.

Returværdi: 0 hvis posten er opdateret/indsat.

Se også: INSERT, REWRITE, DELETE, NOPAS, UPDATE

Eksempel:

```
UPDATE (1)           /* rapporten opdaterer
NOPAS ()             /* intet password
READ(le), #6        /* Læs leverandøren for denne vare
IF #OK THEN BEGIN  /* Hvis denne ikke findes
le#1=#6             /* Indsæt leverandørnummer
le#2="Nyoprettet"   /* Og navn
END
le#6=le#6+#3       /* Opdater leverandørfelter
WRITE (le)          /* indsæt eller opdater leverandør
```

11. Export / Import fra externe filer

Dette afsnit beskriver funktionerne for indlæsning/udlæsning af data i tekstfiler til brug af andre systemer.

11.1. **EXPORT** - Export af data til en tekstfil

tal EXPORT(felter *par1*, filnavn *par2*, tekst *par3*, tekst *par4**6, tekst *par5*, tekst *par6**6)

Beskrivelse: EXPORT exporterer data til en tekstfil. Funktionen kan bl.a. benyttes til flytning af data mellem systemer, til regneark og tekstbehandlingssystemer.

Felterne som angives i *par1* skal angives i som en tekst, dvs. "#1-99" (i gåseøjne).

Det fysiske filnavn som angives i *par2*, såfremt der ikke angives en sti bliver filen placeret i TMP, angives ingen extension bliver denne .OUT, angives slet intet filnavn anvendes rapportens navn, for eksempel c:/tmp/DM1007.OUT for rapport nummer 7.

Med *par3* og *par5* kan man styre recordlængde og linieopdeling af filen.

par4 anvendes normalt kun til filer med fast recordlængde for overførsel til mainframe systemer.

par6 består af 6 karakterer, der bestemmer udseendet af en kommasepareret fil. Bemærk at " i denne streng skal skrives som de to karakterer \". Som standard vil alle alfanumeriske felter skrives som "xxxx", hvor forekomsten af karakteren " (gåseøje) konverteres til ' (ping). De numeriske felter skrives som 99.99, hvor . (punktum) er decimalpunktum. Alle felter adskilles med , (komma).

Export filen kun nu lukkes ved brug af EXPORT("CLOSE"). Dette kan være nødvendigt, hvis man ønsker at se den dannede fil ved at udføre CHAIN til notepad.

Returværdi: Ingen.

Se også: IMPORT

Eksempel:

```
EFTER                                /* EFTER selektion
EXPORT("#1-6","le.csv")              /* alle felter exporteres (CSV)
EXPORT("#1-6","le.csv","","","","--,\"'.") Samme som ovenfor
```

Beregninger og subfunktioner

Ud fra ovenstående eksempel vil filen `le.csv` indeholde følgende:

```
"100","SORENCO A/S","RØDOVREVEJ 273","4711 RØDOVRE"  
"102","DEN DANSKE BANK","LERSØ PARK ALLE 43","2100 KØBENHAVN Ø",,25000  
"105","DANSK OLIE OG GAS","RØDOVREVEJ 373","4711 RØDOVRE",,500
```

Eksempel:

```
EXPORT("#1-6","le.ssv","000001","","","--;. ,") /* alle felter som SSV
```

Ud fra ovenstående eksempel vil filen `le.ssv` indeholde følgende:

```
SW-Tools  
100;SORENCO A/S;RØDOVREVEJ 273;4711 RØDOVRE;;123,25
```

Eksempel:

```
EXPORT("#1-2,5-6","a","-80","1") /* fast længde uden crlf
```

11.2. **IMPORT** - Import af data fra en tekstfil (RAP)

IMPORT(felter *par1*, filnavn *par2*, tekst *par3*, tekst *par4*6*, tekst *par5*, tekst *par6*6*)

Beskrivelse: Funktionen indlæser data fra en tekstfil.

Felterne som angives i *par1* skal angives i "" (gåseøjne). Der kan angives simple beregninger i forbindelse med felterne, for eksempel IMPORT("#1-5,+6","a")

Operator	Funktion
+	Adder til felterne
-	Subtraher fra felterne
&	Skip felterne
=	Sæt felterne lig med
:xx	Skip til position i recorden

Det fysiske filnavn som angives i *par2* kan indeholde en sti, f.eks "c:/export/le.csv".

Returværdi: Ingen.

Se også: EXPORT, IMPOCONT, IMPONEXT, IMPOTHIS

Eksempel:

```

UPDATE (1)                /* rapportens hovedkartotek er le (leverandører)
NOPAS ()                  /* rapporten opdaterer
IMPORT("#1-6","le.csv")   /* hent een post fra (CSV) tekstfil
READ(LE),#1              /* Check om leverandør LE findes
LE#6=LE#6+#6            /* Adder læst beløb til gammel saldo
IF #OK LET("LE#1-6=#1-6") /* Hvis ikke fundet flyt alle felter
WRITE(LE)                /* Opdater/Opret LE leverandør
    
```

Eksempel:

```

IMPORT("#1-6","le.ssv","","","","","--;- -") /* import fra (SSV) tekstfil
    
```

11.2.1. IMPOCONT - Fortsættelse af import (RAP)

IMPOCONT(felter *par1*)

Parametre: *par1* : angiver de felter som skal sættes ved indlæsning af data fra tekstfilen

Beskrivelse: IMPOCONT fortsætter import af yderligere felter fra samme record, som sidste IMPORT og fra den position, hvor denne blev afsluttet. Anvendes for eksempel når en recordtype forrest i recorden kan angive forskelligt feltindhold.

Se også: IMPORT, IMPONEXT, IMPOTHIS

11.2.2. IMPONEXT - Import af næste record (RAP)

IMPONEXT(felter *par1*)

Parametre: *par1* : angiver de felter som skal sættes ved indlæsning af data fra tekstfilen

Beskrivelse: IMPONEXT læser næste record og importerer felter herfra. Anvendes for eksempel når et felt kan angive at der følger een eller flere samhørende records efter denne hovedrecord.

Se også: IMPORT, IMPOCONT, IMPOTHIS

11.2.3. IMPOTHIS - Import af denne record igen (RAP)

IMPOTHIS(felter *par1*)

Parametre: *par1* : angiver de felter som skal sættes ved indlæsning af data fra tekstfilen

Beskrivelse: IMPOTHIS foretager import af samme record igen som sidste IMPORT. Anvendes for eksempel når en recordtype forrest i recorden kan angive forskelligt feltindhold.

Se også: IMPORT, IMPOCONT, IMPONEXT

11.3. FTP - File Transfer Processor filoverførseler

Tal FTP(Tal *par1*, Tekst *par2*)

Par2: FTP kommando

Beskrivelse: FTP funktionen er indbygget for at sætte den avancerede bruger i stand til at over filer for eksempel i fra en rapport baseret på en filliste i en SSV fil. For selve kommandoerne til FTP henviser vi til en FTP manual. Bemærk at frifelter kan anvendes som parameter for kommandoen, samt at 32 bit versionen af TRIO understøtter lange filnavne.

I eksempelet herunder er vist overførsel af et kartotek fra et Quattro system med brug af den specielle kommando QUATTRO for overførsel af hele filen med headerblok, samt XQUAT, der fjerner ekstra information overført fra SSQ serveren på Quattroen for hver blok.

Returværdi: For OPEN: FTP ident, alle andre: FTP fejlkode, 0=OK

Eksempel:

```
#10=FTP(0,"open 200.0.0.9")           /* Opret forbindelse til server
#11=FTP(#10,"user cms mypas")        /* Bruger cms password mypas
#11=FTP(#10,"binary")                /* Start binær overførsel
#11=FTP(#10,"quattro")                /* Start Quattro backup mode
#11=FTP(#10,"get /X.BASIC/0/AFIL c:/mydir/myfil") /* Overfør filen
if #11<>0 FTP(#10,"error")           /* Vis eventuel fejlmeddelelse
#11=FTP(#10,"xquat c:/mydir/myfil")  /* Konverter fra Quattro
#11=FTP(#10,"quit")                  /* Det var det
```

12. Flere firmaer og blanding af kartoteker

De her beskrevne funktioner er beregnet for kørsel med flere firmaer, hvor data for disse er placeret i hver sin database eller tabel, samt til fletning af forskellige kartoteker med samme definition.

12.1. **ACCESS**- Check om en fil findes

tal ACCESS(Filnavn *par1*)

Parametre: *par1* : Filnavn

Beskrivelse: Check om en given fil findes, returner 0 hvis filen fundet.

Returværdi: 0 hvis filen fundet.

Se også: OPEN

Eksempel: IF ACCESS("minfil.ssv")=0 MESS("Ok ? ")

12.2. COMNO - Firmaid

tekst COMNO(kartotek *par1*)

Parametre: *par1* : Blank eller kartoteksid

Beskrivelse: Funktionen returnerer det aktuelle firmaid for det angivne kartotek, for hovedkartoteket hvis intet andet angivet.

Returværdi: Firmaid

Se også: OPCOM

Eksempel: #1 = COMNO() /* Hent nuværende firmaid, fx. "001".

12.3. ENDSUM - Extra grande total ved kørsel med flere hovedkartoteker.

ENDSUM()

Parametre: Ingen.

Beskrivelse: I en rapport med flere selvstændige lister afledt af enten MERGE eller OPCOM funktionen med hver deres totaler kan man yderligere få en total for samtlige udskrevne ved at indlægge ENDSUM() som en beregningslinie.

Systemfelterne #CO og #CN udskrives som *** ved ENDSUM.

Returværdi: Ingen

Se også: KEYS, MERGE, OPCOM

Eksempel: ENDSUM() /* Udskriv extra total til sidst

12.4. FILENAME - Aktuelt filnavn for et kartotek

tekst FILENAME(kartotek *par1*)

Parametre: *par1* : Kartoteksforkortelsen

Beskrivelse: FILENAME returnerer navnet for det kartotek, der er åbnet med den givne id.

Returværdi: Fysisk filnavn.

Se også: OPEN

Eksempel: #1 = FILENAME(va) /* Giver "c:/rapfil/ssv/isa/va.ssv"

12.5. **OPEN** - Åbning af en fil med et givet navn

tal OPEN(kartotek *par1*, Filnavn *par2*, Driver *par3*)

par3 : 0 eller database interface nummer

Beskrivelse: Med denne funktion kan man åbne et bestemt kartotek istedet for det, der allerede er åbnet for den givne kartoteksid. Det tidligere åbne kartotek vil blive lukket.

Der gives fejlmeddelelse på skærmen såfremt kartoteket ikke findes eller af anden grund ikke kan åbnes.

Såfremt *par3* angives vil kartoteket blive åbnet som denne database type som defineret i BASIS.SSV ved database driver installation.

Returværdi: 0=ok, <>0=fejl.

Se også: ACCESS, FILENAME, MERGE, OPCOM

Eksempel:

FØRST

```
OPEN(va, "c:/swtools/demo/va.ssv") /* Anvend dette kartotek
```

```
OPEN(va, #50) /* Indtast kartotek ved start
```


12.5.1. **OPEN** - Temporær lukning af filer

OPEN(filid *par1*, Konstant *par2*)

par2 : "-"

Beskrivelse: En fil kan lukkes for at tillade CHAINede programmer at tilgribe denne. NB: Hovedkartoteket må ikke lukkes med denne funktion.

Returværdi: 0=ok, <>0=error.

Se også: FILENAME, MERGE, OPCOM

Eksempel:

```
OPEN("ku", "-")          /* vil temporært lukke filen for at tillade:
CHAIN("command.com /c edit c:\\windows\\system\\ku.ssv")
```

```
OPEN("ku", "+")          /* vil genåbne filen
```

12.6. **MERGE** - Blanding af flere hovedkartoteker på een rapport (RAP)

tal MERGE(kartotek *par1*, Filnavn *par2*, Driver *par3*)

par3 : 0 eller database interface nummer

Beskrivelse: Med denne funktion kan man sammenstille flere kartoteker på een rapport. Til MERGE kan man enten angive en filid *par1* såfremt filen er defineret som et selvstændigt kartotek eller et direkte filnavn i *par2* som i OPEN. De anvendte kartoteker skal have samme struktur.

Der gives fejlmeddelelse på skærmen såfremt kartoteket ikke findes eller af anden grund ikke kan åbnes.

Såfremt *par3* angives vil kartoteket blive åbnet som denne database type som defineret i BASIS.SSV ved database driver installation.

En rapport, der anvender MERGE, skal normalt være sorteret for at opnå blandingen af kartotekerne for eksempel i varenummer orden. Anvendes MERGE uden sortering fås først en liste fra det normale hovedkartotek og herefter en liste hver af de blandede. ENDSUM funktionen kan anvendes for at få en total af alle udskrevne records.

Kaldes MERGE uden parametre returneres et MERGENUMMER der er 1 for hovedkartoteket, 2 for første blanding, 3 for næste og så fremdeles.

Uden parametre: MERGENUMMER fra 1 og fremefter.

Se også: ENDSUM, OPCOM, OPEN

Eksempel:

```
MERGE (0, "c:/swtools/demo/va.ssv") /* Bland med dette kartotek
MERGE (1e) /* Samt med 1e kartoteket
#12=MERGE () /* Hent blandingsnummer 1,2 eller 3
```

12.7. OPCOM - Åbning af kartoteker i forskellige firmaer

tal OPCOM()

par3 : 0 eller database interface nummer

Beskrivelse: OPCOM funktionen giver tilgang til flere firmaer på en rapport.

En rapport kan bringes til at køre en gang for hvert firma ved at indlægge OPCOM("111,777-888") eller OPCOM(#50) hvor #50 anvendes som start indtastningsfelt. Denne rapport kan herefter udbygges med en total for alle firmaer med ENDSUM eller sorteres for eksempel for at samle informationer om een vare i alle firmaer.

Systemfelterne #CO og #CN kan anvendes til udskrift af firmaid / firmanavn i overskrifterne.

En vareliste defineret på kartotek va kan bringes til at hente informationer fra et andet firmas varekartotek ved at indlægge først OPCOM(VA,"555") og herefter READ(VA) således at va#8 bliver det aktuelle firmas beholdning, VA#7 beholdningen i firma 555.

En rapport baseret på et statistikkartotek med angivelse af firmanummer i statistikrecorden kan åbne de refererede firmaers kartoteker med OPCOM(0,#47)

Såfremt *par3* angives vil kartoteket blive åbnet som denne database type som defineret i BASIS.SSV ved database driver installation.

Kaldes OPCOM uden parametre returneres nuværende firmanummer.

Fra COMPANY.SSV filen hentes firmanavne samt tilladte numre såfremt der er angivet intervaller i firmaid *par2*.

Uden parametre: Firmaid

Se også: COMNO, ENDSUM, MERGE, OPEN

Eksempel:

```
OPCOM("001,777-888")      /* Kør rapporten over disse selskaber
OPCOM("*")                /* Kør rapporten over alle selskaber
OPCOM(va,"123")          /* Anvend varekartoteket selskab 123
OPCOM(0,"777")           /* 777 for alt andet end hovedkartoteket
OPCOM(-1,"888")          /* Anvend firma 888 for alle kartoteker

OPCOM(#50)                /* Indtast firmaer ved start
```

13. DATAMASTER funktioner

Disse funktioner er specielle DATAMASTER og kan ikke anvendes i rapporter. Visse funktioner er også brugbare i IQ, dette vil i så fald være angivet for den enkelte funktion.

13.1. DISABLE- Deaktiver input for et program (IQ)

DISABLE(programnr *par1*)

Parametre: *par1* : Programnr der skal deaktiveres.

Beskrivelse: Deaktiver al input for det givne program nummer.

Returværdi: Ingen.

Se også: ENABLE , FOCUS

Eksempel: DISABLE(20)

13.2. **DISP** - Visning af ændrede felter (IQ)

DISP(Felter *par1*)

Parametre: *par1* : "" eller felter

Beskrivelse: DISP skal anvendes, såfremt man i en beregning for et felt ændrer andre felter, der er placeret på skærbilledet. Hvis DISP udelades, kan man ikke være sikker på, hvornår de nye værdier vil blive vist på skærmen.

DISP() kommandoen til at vise alle felter igen på skærmen er blevet udvidet med at man kan angive kun udvalgte felter skal vises som for eksempel DISP("#1,4")

Returværdi: Ingen

Se også:

Eksempel: DISP()

13.3. DOFUNCTION - Udfør extern funktion (IQ)

DOFUNCTION(Funktion *par1*, text *par2*, programnr *par3*)

par3 : Eventuelt programnr.

Beskrivelse: DOFUNCTION sender en besked <funktionsnr> til det kørende IQ-program eller til det åbne <program>. En nøgle kan medgives til READ funktionerne.

En liste over funktionsnumre kan findes i beregnings-listboxen for 'Beregninger ved valg af en funktion'. For eksempel vil

Returværdi: Ingen.

Se også: CHAIN, PLSNEXT, TRANSMIT

Eksempel:

```
DOFUNCTION(505,#1,20) /* bede program 20 om at læse en record med nøgle #1
DOFUNCTION(550)      /* Zoome ind på dette programs skærbillede
```

13.4. ENABLE- Aktiver input for et program (IQ)

ENABLE(programnr *par1*)

Parametre: *par1* : Programnummer

Beskrivelse: Aktiverer input for et givet programnummer.

Returværdi: Ingen.

Se også: DISABLE , FOCUS

Eksempel: ENABLE(20) /* Aktiver input for program 20

13.5. FOCUS - Aktiver program (IQ)

FOCUS(Programnr *par1*)

Parametre: *par1* : Programnummmmer.

Beskrivelse: Aktiverer input og sætter fokus til det angivne programnummer.

Returværdi: Ingen.

Se også: DISABLE, ENABLE

Eksempel: FOCUS(20) /* Program 20 bliver active

13.6. **FUNC** - Bestem hvilken opdatering, der er nødvendig (IQ)

tal FUNC(kartotek *par1*)

Parametre: *par1* : Kartoteksforkortelse

Beskrivelse: Baseret på brugerens indtastninger bestemmer DATAMASTER om opdatering af en given record er nødvendig, og hvordan denne skal foregå. FUNC anvendes i skriverutinen for at aflæse dette.

Returværdi:

Værdi	Funktion
0	Ingen opdatering nødvendig
1	En bestående record skal modificeres
2	En ny record skal indsættes
3	En bestående record skal slettes

Se også: SETUPD, ON

Eksempel:

```
ON FUNC(cu) GOSUB MAINWRT,MAININS,MAINDEL  
IF FUNC(va) !=3 LET #27=#27+va#4
```

13.7. GETINFO - Hent udvidet program information (IQ/DM)

Tal GETINFO(Tal *par1*, Tekst *par2*)

par2 : Feltreference

Beskrivelse: Med denne funktion kan man få visse specialinformationer fra et IQ/DM program. Type 0 og 1 returnerer en eentydig ident for vinduet som kan anvendes af andre funktioner til opdatering heraf, se eksempel i OLE manualen.

For type 2 til 5 skal der angives en feltreference i *par2*. For at få startkolonnen for varekartotekets felt 7 skal *par2* således være "va#7". Koordinaterne for feltet returneres ud fra den faktiske størrelse som defineret i layoutet. Ønskes i stedet de aktuelle koordinater på skærbilledet med hensyntagen til zoom ind/ud anvendes type 6-9.

Type 2-9 returner en felt koordinat. Værdien kan holdes i et 9,T2 feltformat.

Eksempel:

```
GETINFO(0)           /* Hent IQ programmets vindue ident
GETINFO(2,"va#7");   /* Hent start x koordinaten for va felt 7
```

13.8. HELP - Vis meddelelsesbox med hjælp for et felt (IQ)

HELP(tekst *par1*)

Parametre: *par1* : Feltreference.

Beskrivelse: HELP(#31) Viser en meddelelsesbox med hjælp for det givne felt

Se også: MESS

Eksempel: HELP("#31")

13.9. ISACTIVE - Check om et program er aktivt (IQ)

tal ISACTIVE(Programnr *par1*)

Parametre: *par1* : Programnummer

Beskrivelse: Test om <program> er aktivt.

Returværdi: Returnerer 1 hvis <program> er aktivt, ellers 0.

Se også: CHAIN, EXIT, WAIT

Eksempel: IF ISACTIVE(20)=0 CHAIN(20) /* Start program 20 hvis ikke gjort før

13.10. KEYON - Gem eller vis nøgleinputfeltet (IQ)

KEYON(tal *par1*)

Beskrivelse: KEYON(0) fjerner nøgleinputfeltet, (1) genaktiverer dette.

Returværdi: Ingen.

Se også:

Eksempel: KEYON(0) /* Fjern nøgleinputfeltet

13.11. LINE - Hent eller sæt det aktuelle linienummer (IQ/DM)

Tal `LINE`(Tal *par1*)

Parametre: *par1* : 0, -1 eller linienummer

Beskrivelse: Funktionen vil hente eller sætte det aktuelle linienummer i et IQ/DM program. Dette linienummer er linietaelleren i et program defineret som for eksempel **va#1-6l** eller **le#1-6/va#1-6**.

Hvis *par1* er 0 hentes værdien af linietaelleren.

Hvis *par1* er -1 hentes antal linier defineret. Er programmet defineret med **va#1-6l,t5** returneres således **5**.

Hvis *par1* er større end 0 sættes det aktuelle linienummer til *par1*.

Returværdi: Et linienummer/tæller for *par1* = 0/-1, ellers 0.

Eksempel:

```
#20=LINE() /* Hent det aktuelle linienummer
```

13.12. **LOOP** - Kald en rutine for alle records i liniebufferen (IQ)

LOOP(label *par1*)

Parametre: *par1* : Label (rutinenavnet), der skal kaldes

Beskrivelse: For hver record, der indlæses i et listeprogram, og for hver transaktionslinie i et transaktionsprogram, udfyldes liniebufferen med de læste værdier samt resultatet af beregningerne (ikke-globale workfelter).

Ved skrivning i et sådant program anvendes LOOP til at skrive alle linier, ligesom LOOP anvendes til genberegning af SUM.

Returværdi: Ingen

Se også: GOSUB, ON

Eksempel:

```
LOOP(MAIN)           /* Skrivning i et listeprogram
LOOP(TRANS)          /* Skrivning af transaktionslinier
LOOP(SUMIT)          /* Genberegning af SUM
LOOP(TRANSDEF)       /* Ændring af nøgleværdi for alle transaktioner
```


13.13. MENUCH - Flip menu afchecket flag (IQ)

MENUCH(Menunr *par1*)

Parametre: *par1* : Menunumre

Beskrivelse: Flipper afcheck flag for det givne menupunkt (se MENUS) og opdaterer det tilhørende interne flag for programstyringen.

Returværdi: Ingen.

Se også: MENUUPD, MENUS

Eksempel: MENUCH("31-32") */* Flip IQ opdaterings-menu*

13.14. MENUS - Ændring af menuerne (IQ)

MENUS(Menunr *par1*)

Parametre: *par1* : -xxx=Deaktiver, +xxx=Aktiver menunumre xxx.

Menunummer	Funktion
1/11	Indsæt ny record hovedkartoteket/transaktioner
2/12	Ændring af en record i hovedkartoteket/transaktioner
3/13	Sletning af en record i hovedkartoteket/transaktioner
4/14	Superindex på hovedkartoteket/transaktioner
5/15	Selektioner på hovedkartoteket/transaktioner
6/16	Superindex feltvalg på hovedkartoteket/transaktioner
20	Søgning liste skal match til input
21/22/23/24/25	Transaktioner, Næste/Forrige/Første/Sidste/Retning
26	Vis nøgle ved søgning
27	'Case sensitive' søgning
31/32	Opdater andre/Opdateres af andre
41/42/43/44	Hovedkartoteket, Næste/Forrige/Første/Sidste
51/52/53	Beregninger/Rettelse layout/Save
54/55	Parameter menuer
61/62/63/64	Nyt program, Slet program, Udskriv program, Start program
100-149	Index låst og index menuer
999	Aktiver alt

Beskrivelse: MENUS kan anvendes både i IQ og DATAMASTER programmer til af frakoble ikkeønskede menupunkter.

MENUS kan også aktiveres helt fra start ved kald af IQ fra Windows med -m+xxx eller -m-xxx parametrene. Specielt må man for at kunne rette i beregningerne for et program, hvor denne funktion er deaktiveret, vælge IQ som for eksempel: C:\SWTOOLS\IQWIN -m999

Returværdi: Ingen

Se også: MENUCH, MENUUPD

Eksempel: MENUS("-51-55") /* Deaktiver rettelse af dette program

13.15. MENUUPD - Tilføj til/Kontroller menu (IQ)

MENUUPD(Menunr *par1*, tal *par2*, tal *par3*)

par3 : Tekst

Beskrivelse: Tilføj til / Kontroller menuerne manuelt.

MENUUPD(1,2000,"Min &Egne menu") Tilføj funktion 2000 til menu nummer 1.

Ved valg af dette nye menupunkt vil brugerberegningerne med label FU2000: i 'beregninger ved valg af funktion' blive udført.

Returværdi: Ingen.

Se også: MENUCH, MENUS

Eksempel: MENUUPD(1,2000,"Min &Egne menu") /* Tilføj funktion 2000 til menu nummer 1.

13.16. NEXTFLD - Spring til dette inputfelt (IQ)

NEXTFLD(felt *par1*)

Parametre: *par1* : Det felt, der skal tages input til næste gang.

Beskrivelse: Hvis man ønsker at overstyre den faste feltsekvens afhængig af visse beregninger kan NEXTFLD anvendes til dette.

Brugen af NEXTFLD funktionen er udvidet med programnummer og linienummer.

Returværdi: Ingen

Se også: NEXTFLDSEQ, SEQ

Eksempel:

```
IF #4<#3 NEXTFLD(#3)
NEXTFLD("#10")          /* sætter næste input felt til felt 10
NEXTFLD("#10.2")       /* hopper til felt 10 på linie 2
NEXTFLD("5.#10")      /* hopper til program 5 felt 10
```

13.17. NEXTFLDSEQ - Hop til et inputfelt i en sekvens (IQ)

NEXTFLDSEQ(tal *par1*, tal *par2*)

par2 : Feltnummer i denne sekvens

Beskrivelse: Hop til et bestemt felt i en af feltsekvenserne.

Returværdi: Ingen.

Se også: SEQ , NEXTFLD

Eksempel: NEXTFLDSEQ(2,1) /* Hopper til det første felt i input sekvens 2.

13.18. OBJECTADDSTRING - Adder tekststreng til et objekt (IQ)

OBJECTADDSTRING(felt *par1*, text *par2*, text *par3*)

par3 : Text til brug som index

Beskrivelse: Funktionen indsætter en tekst i et objekt. Resultatet heraf afhænger af objekttypen ifølge nedenstående tabel:

Objekt	Betydning
KNAP	Den tekst, der vises i knappen, indsættes
COMBOBOX	Der tilføjes et nyt element til listen
EDITBOX	Feltteksten i editboxen sættes. Hvis koden for flere edit linier er sat tilføjes teksten efter den bestående
LISTBOX	Der tilføjes et nyt element til listen

Parameteren *par3* bruges kun, hvis objekttypen er COMBOBOX eller LISTBOX. Parameteren skal da indeholde standardværdien for feltet.

Returværdi: Ingen.

Se også: OBJECTCLEAR

Eksempel: OBJECTADDSTRING("va#7",gr#2,gr#1) /* Vis gruppenavn og brug nummer som index

13.19. OBJECTCLEAR - Nulstil indholdet af et objekt (IQ)

OBJECTCLEAR(felt *par1*)

Parametre: *par1* : Felt på skærbilledet, fx. va#7

Beskrivelse: Funktionen nulstiller indholdet af det givne objekt.

Returværdi: Ingen.

Se også: OBJECTADDSTRING

Eksempel:

```
OBJECTCLEAR("va#7")           /* nulstil alle tidligere værdier
START(gr)," "                 /* læs alle værdier fra varagrupperkartoteket
NEXT(gr)
    OBJECTADDSTRING("va#7",gr#2,gr#1) /* Vis gruppenavn og brug nummer som
index
REPEAT(gr)
```

13.20. OBJECTGETSTRING- Hent nummeret på det valgte objekt (IQ/DM)

Tekst OBJECTGETSTRING(Felt *par1*)

Parametre: *par1* : Feltnummer på formen "va#7"

Beskrivelse: Funktionen henter nummeret på den valgte linie i et combobox/ListBox felt, det vil sige den værdi, der svarer til *par3* angivet ved dannelse af boksen med OBJECTADDSTRING.

Funktionen vil typisk finde anvendelse ved klik på et combobox/ListBox felt.

Returværdi: Nummeret på det valgte objekt.

Se også: OBJECTADDSTRING

Eksempel:

```
#20=OBJECTGETSTRING("va#6") /* Hent det valgte leverandørnummer
```


13.21. PLSNEXT - Forbered og læs hovedkartoteket (IQ)

PLSNEXT(tal *par1*, text *par2*, tal *par3*,)

Beskrivelse: Forbered og udfør læsning af hovedkartoteket ifølge den givne mode. Druges i menuerne og ved bladrning ned/op mm. Hvis <inputflag> er sat anvendes <nøgle>, ellers er læsningen næste/foregående eller direkte.

Returværdi: Ingen.

Se også: DOFUNCTION, TRANSMIT

Eksempel: PLSNEXT(0,#1,1) /* læser den næste record med #1 som nøgle.

13.22. **SEQ** - Ændring af inputsekvensen (IQ)

SEQ(tal *par1*, felter *par2*)

par2 : Feltangivelse for den nye sekvens

Beskrivelse: Parameteren for feltsekvens ændres ved hjælp af denne funktion.

Returværdi: Ingen

Se også: NEXTFLD, NEXTFLDSEQ

Eksempel:

```
SEQ(2, "va#2-3,5")           /* Sæt den normale ændringssekvens
IF #7=1 SEQ(2, "va#4,3")     /* Specielt for denne varegruppe
```

13.23. SETUPD - Marker et kartotek på en linie for opdatering (IQ)

SETUPD(kartotek *par1*)

Parametre: *par1* : Kartoteksid der skal opdateres

Beskrivelse: Når der ændres i 'kritiske' felter i hovedkartoteket er det nødvendigt at markere alle transaktioner for opdatering, idet ellers kun poster ændret ved input vil blive skrevet.

Returværdi: Ingen

Se også: LOOP

Eksempel: SETUPD(va)

13.24. **SHOW**- Aktiver/Deaktiver/Vis/Gem et felt (IQ/DM)

Tal SHOW(Felt *par1*, Tal *par2*)

3 = Gem feltet

Beskrivelse: Med denne funktion kan man aktivere/deaktivere input for et felt samt vise/skjule feltet på skærbilledet.

Returværdi: Ingen.

Eksempel:

```
SHOW("va#7",1)          /* Deaktiver felt va#7
```

13.25. **SUPER** - Forbered superindex søgning (IQ)

SUPER(filid *par1*) , tekst *par2*

par2 : Nøgle

Beskrivelse: SUPER funktionen initialiserer NEXT læsningen til at bruge superindex

Returværdi: Ingen.

Se også: NEXT, START

Eksempel:

```
SUPER(va) , #21      /* NEXT anvender superindex søgning for teksten i #21
NEXT(va)            /* skal følge herefter for at læse recorden
SUPER(va)          /* Superindex frakoples igen
SUPER(va) , "#1-3" /* Superindex felter sættes til felt 1-3
```

13.26. TRANSMIT- Opdater andre IQ programmer (IQ)

TRANSMIT(tal *par1*, text *par2*, text *par3*)

par3 : Eventuel forbindelse

Beskrivelse: Send oplysninger om de nuværende records til eet eller flere programmer med anvendelse af de fast definerede forbindelser eller, hvis givet, den angivne forbindelse.

Progid="" Send til alle andre

"20" Opdater kun program 20 (hvis dette er åbent)

"le" Opdater alle programmer, der anvender le som hovedkartotek

Forbindelse = "" Anvend de automatiske forbindelse mellem kartotekerne

"1,2P" Brug felt 1 og 2 pakket som forbindelse

"va.01.6" Brug filen va som udgangspunkt for opdatering

Læs det andet programs hovedfil index 1 med va#6 som

nøgle

Returværdi: Ingen.

Se også: PLSNEXT, DOFUNCTION

Eksempel: TRANSMIT(0,"","") /* Opdaterer alle andre programmer ved hjælp af standard forbindelsen

13.27. TRANSSEL- Definer IQ transaktions selektioner (IQ)

TRANSEL(*tekst par1*, *tal par2*)

Beskrivelse: Skan et eventuelt givet input og definer transaktionsselektioner såfremt dette indeholder en formel som for eksempel #15>0. Bruges af IQ selv i forbindelse med pile op/ned i nøglefeltet.

Returværdi: Ingen.

Se også:

Eksempel: TRANSSEL("#15>20",1) */* Definer selektion*

14. SYSTEM funktioner

Disse funktioner er beregnet til specielle programmer, hvor man for eksempel ønsker direkte tilgang til filer / stier.

14.1. DEBUG- Aktiver debug vindue (IQ)

DEBUG(felter *par1*)

Beskrivelse: DEBUG(1) åbner et vindue som viser alle beregninger og deres programnummer/label når disse udføres.
DEBUG vinduet lukkes når IQ lukkes.

Returværdi: Ingen

Se også: WIF, WIFS

Eksempel: DEBUG(1) /* Aktiver debug vinduet

14.2. **EXEC**- Udfør en tekststreng som en beregning

EXEC(*tekst par1*, Programnr. *par2*)

par2 : **(IQ/DM)**Eventuelt program nummer

Beskrivelse:

```
#20="#2=17"
```

```
EXEC(#20)
```

udfører en tekststreng i felt 20 som en beregning.

Hvis man anvender frifelter i EXEC funktionen skal man anvende WW#nn referencen for feltet, denne fås ved udskrift af programdokumentationen.

Generelt gælder, at den streng, der overgives til EXEC funktionen ikke pre-skannes og checkes som en normal beregningslinie. Dette har specielt betydning ved anvendelse i RAPGEN idet C-syntaksen for beregninger skal følges. Vi tilråder bestemt ikke-programmører til at holde brugen af EXEC funktionen i RAPGEN så simpel som mulig uden at involvere andre funktionskald. Forkerte funktionsparametre kan medføre Windows generel beskyttelsesfejl.

Specielt et punkt skal man være opmærksom på i RAPGEN: #15=2 sætter felt 15 lig med 2 OGSÅ når man bruger dette udtryk som IF #15=2 LET #16=3. Man skal angive lighedstegnet dobbelt ifølge C-syntaksen i en sådan beregning: IF (#15==2) LET #16=3

IQ: EXEC(#20,15) skifter til det aktive program 15 og udfører den angivne beregning.

Returværdi: Ingen

Se også:

Eksempel: EXEC(#20) /* Udfør en beregning, der indtastes ved start af rapporten

14.3. GETFLD- Sæt SY strukture pointers (IQ)

GETFLD(tekst *par1*)

Parametre: *par1* : Feltspecification

Beskrivelse: Denne funktion sætter system variable (SY#..) til at pege på definitionen for det givne felt. Feltdefinitionen kan herefter aflæses eller ændres under kørselen. Kun til speciel programmørbrug.

Returværdi: Ingen

14.4. INSTALL- Externe funktioner

INSTALL(text *par1*, text *par2*, text *par3*, text *par4*)

par4 : Eventuelt eget funktionsnavn

Beskrivelse: Programmører med kendskab til funktionsdefinitioner fra andre DLL'er kan inkludere disse som IQ funktioner.

NB: Forkert brug af denne funktion kan forårsage systemnedbrud.

Returværdi: Ingen

Se også:

Eksempel:

```
INSTALL("a.dll","b","3,[ss]")
```

aktiverer #20=B(#21) fra a.dll, #20 og #21 er short variable

```
INSTALL("mit.dll","etnavn","3,[sCl]","FUNNY")
```

aktiverer #30=FUNNY(#31,#32) som funktionen etnavn fra mit.dll

returværdien #30 short, parametre #31 som char pointer, #32 som long.

14.5. SYSPAR - Hent systemparameter

tekst SYSPAR(tal *par1*)

Beskrivelse: SYSPAR henter den angivne systemparameter, hvoraf kun de ovenførte numre er relevante.

Returværdi: Systemparameteren

Se også: SYSPARSET

Eksempel: #1 = SYSPAR(4) /* Hent TMP stien

14.6. SYSPARSET - Sæt værdien af en systemparameter

SYSPARSET(tal *par1*, tekst *par2*)

par2 : Ny værdi af denne system parameter

Beskrivelse: SYSPARSET sætter en ny værdi for den angivne systemparameter.

Returværdi: Ingen.

Se også: SYSPAR

Eksempel: SYSPARSET(4,"c:/mintmp/") /* Sæt ny TMP sti

14.7. USERINFO - Hent brugerinformation fra brugerstyringen

Tekst USERINFO(Tal *par1*)

17=Brugerdefineret

Beskrivelse: Denne funktion henter information fra brugerstyringen om den aktive bruger. Nummeret angiver i *Par1* refererer til feltnummeret i systemfilen US, hvor man selv kan definere felt 11-17 individuelt for installationen (Pas i så fald dog på ved eventuel senere opgradering af TRIO).

Returværdi: Tekststreng med den ønskede brugerinformation.

Eksempel:

```
#11=USERINFO(6) /* Hent det første bemærkningsfelt fra brugeren
```

14.8. WIF - Testprint (IQ)

WIF(*tekst par1*)

Parametre: *par1* : Tekst

Beskrivelse: WIF giver et testprint til filen c:/wif uden af forstyrre skærbilledet.

Returværdi: Ingen

Se også: WIFS, DEBUG

Eksempel: WIF("Her er jeg") /* *Testprint af en tekst*

14.9. WIF- Testprint (RAP)

WIF(*tekst par1* , *tekst par2*)

. **Beskrivelse:** WIF giver et testprint til filen c:/wif.

Returværdi: Ingen

Se også: WIFS, DEBUG

Eksempel: WIF("Feltet er %s.",#2) /* Testprint

14.10. WIFS- Testprint af felter (IQ)

WIFS(felter *par1*)

Parametre: *par1* : Feltspecifikation

Beskrivelse: WIFS giver et testprint til filen c:/wif af de angivne felter.

Returværdi: Ingen

Se også: WIF, DEBUG

Eksempel: WIFS("va#1-3,le#2") /* Testprint af feltværdier

Index

A

ABS32
Afrunding 33;34;39

B

Billedefelter30
Blok-sætninger 3;6
Brugernavn.....24

C

CCODE 60;63;64
CHAIN
 95;96;97;98;99;101;102;145;158;164;170
CHECK61;62
CHEX61;62
CLEAR..... 84;87;94;141
COLOR89;90
COLORF89;90
COMNO 154;160
COMPANY 160
COMPILE 100
CONV.....44;48;53

D

DATAMASTER
 28;53;60;95;96;139;161;167;175
DATE66
DATECALC
 67;68;69;70;71;72;75;76;78;80;82
Datoberegning65
Datovending71
DAY 67;68;69;70;71;72;75;76;78
DELETE 135;139;140;141;142;143
DISP..... 163

E

EDIT 45;50;58
ENDSUM.....103;124;155;159;160
EXIT 96;97;98;99;101;102;107;170
EXPORT.....145;146;147

F

FILENAME..... 27;156;157;158
FIND46
Firmaid 154;160
FNA67;68;69;70;71;72;73;75;76;78;80;82
FNB 67;68;69;70;72;75;76;78;80;82
FND
 .67;68;69;70;71;72;74;75;76;77;78;80;
 82
FNE72
FNF.....73
FNH 33;34;35;39
FNO 71;74;77
FNR 33;34;35;39;40

FNU 67;68;69;70;71;75;76;78;80;82
FNV67;68;69;70;71;72;75;76;78;80;81;82
FNY 67;71;74;77
FRA 35;36;103
FUNC 167

G

GETKEY 134

I

IMPOCONT..... 147;148;149;150
IMPONEXT 147;148;149;150
IMPORT 145;147;148;149;150
IMPOTHIS..... 147;148;149;150
INDEX 103;104
INSERT 86;87;139;140;141;142;143
INT 36;39;41
IQ
 28;85;88;91;92;93;95;96;98;99;102;12
 5;161;162;163;164;165;166;167;168;16
 9;170;171;172;173;174;175;176;177;17
 8;179;180;181;182;183;184;185;186;18
 7;188;190;191;192;193;197;199

K

KEYS.....103;104;155
Kvadratoden 42

L

LEN47;55
LOOP 173;184
LOWER..... 44;48;53;54;57
LTOT..... 105;106

M

MENUS.....174;175;176
MERGE 112;155;157;158;159;160
MESS95;101;102;107;153;169
MONTH67;68;69;70;71;72;75;76;78;80;82
MTOT 105;106

N

NAME.....49;53;54
NEXT
 7;126;127;131;132;133;135;136;180;18
 6
NEXTFLD177;178;183
NOPAS 108;109;139;140;141;142;143;147
NUMBER..... 18;27;45;50;51
NUMS 18;50;51

O

OCR 61
OPCOM..... 154;155;157;158;159;160
OPEN 151;153;156;157;158;159;160

P

PACK52;56

Beregninger og subfunktioner

PAGE 117;118
Pakning52
PAS 108;109
Potensopløftning.....38
POW38;42
PRINT
 6;8;10;12;17;117;118;119;120;121;122
 ;123;125;132;136
PRIOR..... 127;131;132;133;135;136
PRTTOTAL 124
R
RAPDAY.....82
RAPGEN
 26;28;29;95;100;113;117;126;191
READH 118;128
READR127;129;130
READX127;129;130
REWRITE.....139;140;141;142;143
RUN..... 33;34;35;39
RUND.....34;40
S
SEQ177;178;183
SETUPD..... 167;184
SGN..... 32;37;41
SIDST 97;124

SMAA..... 44;48;49;53;54;57
SORTD113
SORTKEY.....112
SORTWORK113
SPOFF47;55
SQR.....38;42
U
Udpakning 56
UNPACK 52;56
UPDATE
 86;87;108;135;139;140;141;142;143;14
 7
UPPER..... 44;48;53;54;57
USING..... 18;45;50;58
W
WDAY .67;68;69;70;71;72;75;76;78;80;82
WEEK.....76;81
WORDS49;53
WORKD
 67;68;69;70;71;72;75;76;78;80;82
WRITE..... 139;140;141;142;143;147
Z
ZERO 84;87;94