Norsk Data

NOTIS TF — Makro
Veiledning

ND-63.010.01

NOTIS TF — Makro
Veiledning

ND-63.010.01

Kunngjøring

Opplysningene i dette dokumentet kan endres uten varsel. Norsk Data er ikke ansvarlig for feil som måtte forekomme i dette dokumentet. Norsk Data er ikke ansvarlig for sine bruker-systemers anvendelse eller pålitelighet på maskiner som ikke er levert eller anbefalt av Norsk Data.

De opplysninger som er gitt i dette dokumentet er beskyttet av opphavsrett. De må ikke fotostat-kopieres, reproduseres eller oversettes uten at Norsk Data's tillatelse først er innhentet.

Copyright © 1983 — Norsk Data A.S.

TRYKKINGSFORTEGNELSE

Utgitt	Kommentarer
08/83	Versjon 01

Publ.nr. ND-63.010.01
NOTIS-TF Makro Veiledning
August 1983

Norsk Data

Norsk Data A.S
Grafisk Senter
Postboks 25, Bogerud
0621 Oslo 6, Norge

Håndbøker kan oppdateres på to måter: nye utgaver og revisjoner. Nye utgaver består av en helt ny håndbok som erstatter den gamle håndboken. Nye utgaver inneholder alle revisjoner som er foretatt siden den forrige utgaven ble sendt ut. Revisjoner består av en eller flere enkeltsider som brukeren selv skal sette inn i håndboken. Alle reviderte sider oppgis på en ny trykkingsfortegnelse som sendes ut sammen med revisjonen. Denne nye trykkingsfortegnelsen skal erstatte den gamle.

Nye utgaver og revisjoner annonseres i ND Bulletinen, og kan bestilles som forklart nedenfor.

Leserens kommentar-ark bakerst i denne håndboken kan brukes både til å informere om feil i håndboken, og til å gi en bedømmelse av denne. Både detaljerte og generelle kommentarer er velkommen.

Arkene, samt forespørsler om dokumentasjon sendes til:

Dokumentasjonsavdelingen
Norsk Data A.S
Boks 25, Bogerud
0621 Oslo 6

Bestillinger av dokumentasjon sendes til det lokale ND-kontoret eller (i Norge) til:

Grafisk Senter
Norsk Data A.S
Boks 25, Bogerud
0621 Oslo 6

Forord:

PRODUKTET

Denne veiledningen er ment å være en total dokumentasjon av bruk av makroer i Norsk Datas tekstformateringssystem

NOTIS-TF SUT 10079J

LESEREN

Veiledningen er først og fremst beregnet for erfarne brukere av tekstbehandlingssystemet NOTIS-WP og tekstformateringssystemet NOTIS-TF. Håndboken vil være av spesiell interesse for brukere som vil lage sine egne makroer tilpasset egne behov for spesielle dokumentformater.

FORHÅNDSKUNNSKAPER

Det forutsettes at brukeren har en viss erfaring i bruk av tekstformateringssystemet NOTIS-TF, og kjenner til bruken av direktiver og dokumentmakroer som er beskrevet i NOTIS-TF Håndbok - Tekstformaterer.

VEILEDNINGEN

Veiledningen gir en teoretisk beskrivelse av oppbygging og virkemåte av de forskjellige typer makroer som finnes i NOTIS-TF. Veiledningen forklarer hvordan disse kan defineres, endres, slettes og brukes. Det tatt med mange eksempler som viser hvordan de virker i praksis. Kapittel 10 er i sin helhet viet til forklaring av eksempler, der de fleste eksemplene er hentet fra det offisielle makro-biblioteket i NOTIS-TF.

ANDRE AKTUELLE HÅNDBØKER

NOTIS-TF Håndbok - Tekstformaterer, ND-63.008, beskriver standard- og avanserte direktiver i tekstformateringssystemet NOTIS-TF.

NOTIS-WP Håndbok - Editor, ND-63.004, tar for seg tekstbehandlingssystemet NOTIS-WP.

INNHOLDSFORTEGNELSE

Seksjon — Side

1 INNLEDNING 1

1.1 Hva er en makro? 1
1.2 Når kan du bruke makroer? 1
1.3 Definering og kalling av makroer 2
1.4 Ordliste 3
1.5 Forskjellige typer makroer 4
1.5.1 Bruker-makroer (MD) 5
1.5.2 Integer-makroer (IM) 6
1.5.3 System-makroer 7
1.5.4 Referanse-makroer (RD) 8
1.5.5 Trigger-makroer (TM,ST) 9
1.5.6 Trigger-strenger 10

2 FORKLARING TIL SYNTAKS 11

2.1 Generell syntaks for makro-kall 11
2.2 Kall-nivåer 12
2.3 Sitattegn ("Directive Quotes") 14

3 INNGÅENDE BESKRIVELSE AV MAKRO-TYPEN SYSTEM-MAKRO 17

4 INNGÅENDE BESKRIVELSE AV BRUKER-MAKROER 20

4.1 MD, Makro define 21
4.1.1 Brukermakro med konstant stamme 22
4.1.2 Brukermakro med makro-kall i stammen 22
4.1.3 Brukermakro med variabel del i stammen - makro-kall i sitattegn 23
4.1.4 Brukermakro med parameterinkludering 24
4.1.5 Parameterinkludering med standardverdier 25
4.1.6 Parameterinkludering ved hjelp av direktivet PM 27
4.2 MK, makro kill 29
4.3 MA, makro append 30
4.4 MI, makro insert 31

Seksjon Side

4.5 MR, makro remove 32
4.6 CM, clear makro 33
4.7 DM, dump makro 34
4.8 FIX, Fiksér makro 35
4.9 NREDEF, Set No-redefine 36

5 INNGÅENDE BESKRIVELSE AV MAKRO-TYPEN REFERANSE-MAKRO 37

5.1 RD, reference define 37
5.2 MK, makro kill 38
5.3 MA, makro append 38
5.4 MI, makro insert 39
5.5 MR, makro remove 39
5.6 CM, clear makro 39
5.7 DM, dump makro 39
5.8 FIX, fikser makro 39
5.9 NREDEF, set no-redefine 39

6 INNGÅENDE BESKRIVELSE AV MAKRO-TYPEN INTEGER-MAKRO 41

6.1 IM, integermakro define 41
6.2 MK, makro kill 42
6.3 DM, dump makro 42
6.4 FIX, fiksér makro 42
6.5 NREDEF, set no redefine 42

7 INNGÅENDE BESKRIVELSE AV MAKRO-TYPEN TRIGGER-MAKRO 43

7.1 TM, trigger-makro define 43
7.2 MK, makro kill 46
7.3 MA, makro append 46
7.4 MI, makro insert 46
7.5 MR, makro remove 46
7.6 DM, dump makro 46
7.7 FIX, fiksér makro 46
7.8 DISABLE, setter en trigger-makro passiv 47
7.9 ENABLE, setter en trigger-makro aktiv 47
7.10 PRIOR, setter prioritet på en trigger (variabel) 47
7.11 Single-triggers 48

8 INNGÅENDE BESKRIVELSE AV MAKRO-TYPEN TRIGGER-STRENGER . . . 49

Seksjon Side

8.1 TP-STR, Titlepage string (Tittelside-streng) 50
8.2 PH-STR, Pageheader string (Sidehode-streng) 50
8.3 TL-STR, Trailer string (Sidehale-streng) 50
8.4 EOF-STR, End of file string 51
8.5 EOD-STR, End of document string 51
8.6 PAR-STR, Paragraph string (Avsnitts-streng) 51
8.7 UND-STR, Underline string (Understrekings-streng) 52

9 SPESIELLE DIREKTIVER SOM ER MYE BRUKT I MAKRO-HÅNDTERING . . 53

9.1 Direktivene OF og NF 53
9.2 Direktivet AR 54
9.3 Direktivet IF 56
9.4 Bruk av makro-biblioteker med direktivet LIB-CONT 60
9.5 NOTIS-TF - ressurs- og plassforbruk i maskinen 62

10 EKSEMPLER 65

10.1 Makroen "BOLD" 66
10.2 Makroen "UTHEV" 67
10.3 Eksempel-makroen i denne veiledningen 68
10.4 Makroen "BM" 68
10.5 Makroen "CE" 69
10.6 Makroen "CP" 70
10.7 Makroen "SPRE" 71
10.8 Makroen "FIG-CP" 73
10.9 Makroen "SIP" 82
10.10 Makroen "HODE" 93
10.11 Makroen "NOTAT" 96

EKSEMPELOVERSIKT 99

Stikkord 101

1 INNLEDNING

1.1 HVA ER EN MAKRO?

En makro er et sammensetning av kommandoer/direktiver som kan defineres for å gjøre trivielle og tidskrevende rutiner enklere. Hver gang du skal utføre en bestemt rutine skriver du inn navnet på den definerte makroen i tekstdokumentet, og den *kalles/ekspanderes* under formateringen, og utfører rutinen automatisk. Makroen gis en unik identifikasjon (navn). Definisjonen skrives enten inn i tekstdokumentet, eller inkluderes i et eget makro-bibliotek. NOTIS-TF har et eget standard makro-bibliotek som utfører mange oppgaver.

Forskjellen mellom en makro og et direktiv i NOTIS-TF, er at de fleste makroer kan defineres, endres og slettes av brukeren selv, slik at man kan bygge opp et spesialtilpasset bibliotek av slike makroer. Vanlige direktiver er fast definert i systemet, og kan ikke forandres av brukeren. I bruk er de to typene identiske. Det vil si at de skrives inn i tekstdokumentet, og utføres (ekspanderes) når dokumentet formateres.

En makro utfører ofte arbeidet til en hel rekke direktiver, og kan brukes til mange forskjellige ting.

1.2 NÅR KAN DU BRUKE MAKROER?

Makroer kan brukes i en rekke sammenhenger. Her nevner vi kun noen få eksempler, og overlater til din egen fantasi og behov å finne andre anvendelsesområder.

1) Dersom du har en bestemt tekstbit, gjerne et langt ord eller uttrykk, som går igjen gjennom hele teksten, kan du definere en makro for dette uttrykket. Du kaller da denne makroen hver gang du skal ha med uttrykket, og du sparer tid og er sikker på at det blir korrekt skrevet hver gang.

2) Dersom du har en spesiell sekvens av kommandoer og direktiver og/eller tekst som forekommer mange steder gjennom hele dokumentet, kan du definere en makro som utfører denne sekvensen. Dette kan redusere skrivearbeidet en god del.

3) Du kan bruke makroer for å lette oppdatering av tekstbiter som forandres gjennom et helt dokument, f.eks. versjonsnummer i program-dokumentasjon.

4) Makroer kan brukes til å beskrive formater som brukes flere steder i dokumentet eller i en rekke dokumenter. Dette gjør at du ved kun å forandre makro-definisjonen, kan forandre på tekst-layouten alle steder der dette direktivet er brukt. Denne bruken av makroer letter bruken av standard-formater.

5) Du kan bruke makroer for automatisk nummerering av tekstbiter, f.eks. avsnitt. Dette gjør at du kan sette inn nye tekstbiter uten å måtte nummerere om de eksisterende manuelt. Dette er gjort med makroene som styrer kapittel- og seksjonsinndeling i NOTIS-TF.

6) Makroer kan brukes for å lage referanser i et dokument. Dette gjør at du kan forandre på dokumentet og likevel opprettholde riktig referansehenvisning.

Dette er bare noen få enkle eksempler på hva makroer kan brukes til. Vi vil komme tilbake til mer konkrete eksempler senere.

1.3 DEFINERING OG KALLING AV MAKROER

Det er viktig å forstå forskjellen mellom en makro-definisjon og et makro-kall:

- En *makro-definisjon* er en spesifikasjon av hva som skal skje hver gang makroen kalles. Den forekommer derfor bare én gang, og skrives inn enten som en del av tekstdokumentet, eller inkluderes som en del av et makro-bibliotek.

- Et *makro-kall* skriver du inn i teksten din hver gang du vil utføre den spesifiserte makro-definisjonen. Et makro-kall kan derfor forekomme mange ganger i teksten. Under formatering av dokumentet vil NOTIS-TF utføre eller *ekspandere* makro-kallet i henhold til definisjonen.

La oss illustrere det med et lite eksempel:

Eks-1: *Du holder på å skrive en lengre rapport om* **acetylsalicylsyre**. *Dette er et langt og vanskelig ord, og det er lett å gjøre skrivefeil når du må skrive det om og om igjen hundrevis av ganger gjennom hele rapporten. Da definerer du heller en makro for dette ordet. Forrest i dokumentet skriver du da inn makro-definisjonen:*

^md/ace/acetylsalicylsyre;

Her har du definert en makro med navnet **ace**. *Hver gang du så får bruk for å ha ordet* **acetylsalicylsyre** *i teksten, kaller du makroen ved å skrive inn* **^ace;**, *f.eks slik:*

Det er ingen tvil om at ^ace;ns egenskaper kan være svært nyttige...

Etter at du har formatert dokumentet ditt, blir da resultatet slik:

Det er ingen tvil om at acetylsalicylsyrens egenskaper kan være svært nyttige...

Legg merke til at vi måtte legge til **ns** *etter makro-kallet, siden makroen bare var definert som* **acetylsalicylsyre**, *mens vi ville ha* **acetylsalicylsyrens**.

Mange makroer er definert med *parametre*, på samme måte som vanlige direktiver kan ha parametre. En parameter er en verdi som du må gi i tillegg til navnet når makroen kalles.

Som du vil se senere i veiledningen, kan mange makro-definisjoner være ganske lange og gå over flere linjer. Det er da en god regel at alle nye linjer begynner med en hatt, '^'. Da vil eventuelle blanke forrest på linjen bli ignorert og vil ikke skape problemer med uventede resultater.

1.4 ORDLISTE

Under følger en liste over ord og uttrykk med definisjoner som er mye brukt i veiledningen. Den kan du bla tilbake tilbake til når du støter på ett av uttrykkene i teksten, og ikke husker definisjonen.

makro	En definert sammensetning av kommandoer/direktiver og/eller tekst som gis en unik identifikasjon. En makro kan defineres og modifiseres av brukeren. Den kalles som et vanlig direktiv i NOTIS-TF, og utfører spesifiserte operasjoner.
makro-definisjon	En spesifikasjon av hvilke operasjoner som skal utføres hver gang makroen kalles.
makro-kall	Et uttrykk som skrives inn i teksten hver gang du vil at makroen skal ekspanderes. Har samme format og bruksmåte som et vanlig direktiv i NOTIS-TF.
makro-ekspansjon	Den prosess som et makro-kall resulterer i når dokumentet formateres. En makro-ekspansjon produserer et resultat i henhold til makro-definisjonen.
nivå	Ekspandering av en makro kan foregå på flere nivåer. Dette er tilfelle hvis et makro-kall inneholder kall til andre makroer, eventuelt dersom sitattegn er brukt.

sitattegn	Sitattegn brukes til å hindre ekspansjon av makro-kall, og bestemmer også hvilket nivå makroekspansjonen til enhver tid skal foregå på.
bruker-makro	En makro-type som typisk brukes til å definere spesielle formater og layout på teksten.
system-makro	En forhåndsdefinert makro-type i systemet. Kan ikke påvirkes av brukeren. Brukes til å få tilgang til faste verdier i systemet.
integer-makro	En makro-type som inneholder en heltallsverdi eller et aritmetisk uttrykk. Brukes typisk til å telle på forskjellige måter, f.eks. holde orden på innbyrdes nummerering av kapitler og seksjoner.
referanse-makro	En makro-type som brukes til referanser/-henvisninger innen et dokument.
trigger-makro	En makro-type som bare kan defineres, ikke kalles. Den ekspanderer seg selv hver gang de definerte vilkår tilfredsstilles.
Parameter	Makroer kan defineres til å inneholde parametre. Disse må da inkluderes i makro-kallet. Antall parametre i kallet må stemme overens med det definerte antall i definisjonen.

1.5 FORSKJELLIGE TYPER MAKROER

NOTIS-TF opererer med 6 forskjellige typer makroer:

	Definisjon	Kall
BRUKER-MAKROER	^MD/NAVN/STAMME;	^NAVN/PARAMETER;
INTEGER-MAKROER	^IM/NAVN/VERDI;	^$NAVN/TYPE;
SYSTEM-MAKROER		^$NAVN/TYPE;
REFERANSE-MAKROER	^RD/NAVN/STAMME;	^#NAVN/STØRRELSE;
TRIGGER-MAKROER	^TM/NAVN/VILKÅR/STAMME;	
	^ST/NAVN/VILKÅR/STAMME;	
TRIGGER-STRENGER	^NAVN/STAMME;	

1.5.1 BRUKER-MAKROER (MD)

Bruker-makroer er den vanligste typen makroer. De brukes typisk til å definere spesielle formater og layout på teksten. Makroen i eksemplet på side 2 er en bruker-makro. Bruker-makroer kan defineres til å inneholde en sammensetning av direktiver og/eller tekst.

Bruker-makroen består av to deler: *Navn* og *Stamme*, og defineres på følgende måte:

`MD/Navn/Stamme;

Navn kan bestå av en kombinasjon av bokstaver, tall og tegnet '-' (bindestrek). Men et navn må alltid begynne med en bokstav, og slutte med en bokstav eller et tall.

Stammen kan bestå av en kombinasjon av tekst og direktiver/makroer.

Makroen kalles/ekspanderes på følgende måte:

`Navn/parameter-1/parameter-2/... ../parameter-n;

Parametrene er valgfrie verdier som du må skrive inn sammen med navnet. Antall parametre du eventuelt må gi sammen med makronavnet er definert i stammen i makro-definisjonen. Ofte er ikke makroer definert med parametre i det hele tatt. Det var tilfellet med makroen *ace* på side 2.

Kapittel 4 gir en utfyllende forklaring av brukermakroer.

1.5.2 INTEGER-MAKROER (IM)

Integer-makroen er en makro som inneholder en heltallsverdi som kan ligge i området -32768 til 32767. Denne typen makro brukes typisk til å telle på forskjellige måter, og lagre tall i makroer som igjen kan kalles fra andre makroer. Nummereringen av eksempler i denne veiledningen er et eksempel der integer-makroer er brukt.

Integer-makroen består av to deler: *Navn* og *Verdi*, og defineres på følgende måte:

^IM/Navn/Verdi;

For *Navn* gjelder samme regler som for bruker-makroen.

Verdi kan være en heltallsverdi eller et aritmetisk uttrykk bestående av heltallsverdier.

Makroen kalles/ekspanderes på følgende måte:

^$Navn/Type;

Type kan ha verdiene:

A : Alfabetisk representasjon med store bokstaver.

a : Alfabetisk representasjon med små bokstaver.

R : Representert i romertall med store bokstaver.

r : Representert i romertall med små bokstaver.

N : Numerisk representasjon.

n : Numerisk representasjon.

Kapittel 6 gir en utfyllende forklaring av integer-makroer.

1.5.3 SYSTEM-MAKROER

System-makroer er en type makroer som er forhåndsdefinert i systemet og som ikke kan påvirkes av brukeren. Disse brukes til å få tilgang til en rekke såkalte systemverdier, som f.eks. dagens dato, aktuelt sidenummer, linjenummer, kapittelnummer, osv.

Makroen kalles/ekspanderes på følgende måte:

^$Navn/Type;

Type kan ha verdiene:

A : Alfabetisk representasjon med store bokstaver.

a : Alfabetisk representasjon med små bokstaver.

R : Representert i romertall med store bokstaver.

r : Representert i romertall med små bokstaver.

N : Numerisk representasjon.

n : Numerisk representasjon.

Kapittel 3 gir en utfyllende forklaring av system-makroer.

1.5.4 REFERANSE-MAKROER (RD)

Referanse-makroer kan sammenlignes med bruker-makroer, med den forskjell at referanse-makroen kan kalles før den er definert. De kan brukes til referanser/henvisninger innen et dokument, også henvisninger fremover til deler av dokumentet som ennå ikke er skrevet.

Referanse-makroen består av to deler: *Navn* og *Stamme*, og defineres på følgende måte:

^RD/Navn/Stamme;

For *Navn* og *Stamme* gjelder samme regler som for bruker-makroer.

Referanse-makroen kalles/ekpanderes på følgende måte:

^#Navn/størrelse;

Størrelse er den lengden/størrelsen som kreves for å få plass til den teksten som blir resultatet av den ekspanderte makroen. Denne størrelsen er strengt tatt bare påkrevd i de tilfeller hvor makroen kalles før den er definert, men det er en god regel å alltid ha den med. I de tilfeller hvor den avsatte størrelsen viser seg å være større en nødvendig, vil teksten bli høyrejustert i det avsatte feltet. I de tilfeller hvor den avsatte størrelse er for liten, vil den aktuelle linjen bli utvidet med det nødvendige antall posisjoner ut i høyre marg, og det vil blitt gitt beskjed under formateringen.

Kapittel 5 gir en utfyllende forklaring av referanse-makroer.

1.5.5 TRIGGER-MAKROER (TM.ST)

Trigger-makroen er en komplisert form for bruker-makro, og er ikke alltid like lett å forstå og bruke.

En trigger-makro består av fire deler: *Navn*, *Vilkår*, *Stamme* og *Prioritet*, og defineres på følgende måte:

```
        ^TM/Navn/Vilkår/Stamme/Prioritet;
eller   ^ST/Navn/Vilkår/Stamme/Prioritet;
```

For *Navn* gjelder samme regler som for bruker-makroer.

Vilkår er et logisk uttrykk som er nøyere forklart i seksjon 7.1.

For *stammen* gjelder samme regler som for bruker-makroer, med det unntak at det ikke er mulig å ha med parametre i en trigger-makro.

Prioritet er forklart i seksjon 7.1.

En trigger-makro skal ikke kalles, den ekspanderes automatisk når de definerte vilkår er tilfredsstilt. Kort forklart betyr det at med en gang det oppstår en situasjon som gjør at de/det definerte vilkår tilfredsstilles, ekspanderes makroen, og den ekspanderes om og om igjen inntil de/det definerte vilkår ikke lenger er tilfredsstilt.

Kapittel 7 gir en utfyllende forklaring av trigger-makroer.

Single-triggers er et spesialtilfelle av trigger-makroer. Dette er en makro som ekspanderes på samme måte som trigger-makroen, men den sletter seg selv etter første ekspandering. Det betyr at en single-trigger aldri ekspanderes mer enn én gang.

1.5.6 TRIGGER-STRENGER

Trigger-strenger er en forhåndsdefinert form for single-trigger. De har forhåndsdefinerte navn, vilkår og prioritet, mens stammen kan defineres av brukeren.

Det finnes en system-makro for hver trigger-streng, et *flagg* som forteller om strengen er i bruk eller ikke.

En trigger-streng-stamme defineres på følgende måte:

^NAVN/stamme;

der *navn* er det forhåndsdefinerte navnet til makroen.

For *stamme* gjelder samme regler som for trigger-makroer.

Kapittel 8 gir en utfyllende forklaring av trigger-strenger.

2 FORKLARING TIL SYNTAKS

2.1 GENERELL SYNTAKS FOR MAKRO-KALL

Den generelle kall-syntaksen for makroer er den samme som for vanlige direktiver:

1) Kallet innledes med tegnet for direktiv-start, standard er hatt, '^'.(Dette tegnet kan om ønskelig redefineres til å være et annet ved hjelp av direktivet ^DS.) For integer-makroer følger så tegnet '$', og for referanse-makroer '#'.

2) Så kommer navnet til makroen, etterfulgt av skilletegn. Dette skilletegnet kan være et hvilket som helst ikke-alfabetisk, ikke-numerisk tegn med unntak av '-' (bindestrek) og tegnet for direktiv-slutt. Men det *må* brukes *samme skilletegn* mellom alle parametrene i kallet. Dette betyr at det første ikke-alfabetiske, ikke-numeriske tegnet forkjellig fra '-' og direktiv-slutt blir kallets skilletegn.

3) Så følger eventuelle parametre adskilt med kallets skilletegn. Antall parametre må stemme overens med det definerte antall i definisjonen.

4) Kallet avsluttes med direktiv-slutt-tegnet, standard er semikolon, ';'. (Dette tegnet kan om ønskelig redefineres til å være et annet ved hjelp av direktivet ^DE.)

Legg merke til at dersom du skriver et direktiv-start-tegn etterfulgt av et mellomrom (blank), vil dette direktiv-start-tegnet bli skrevet og mellomrommet fjernet.

Eks-2: ^ *ETTER-NAVN;*

I eksemplet over vil det ikke bli utført noe kall til makroen *ETTER-NAVN*, men det vil bli stående ^*ETTER-NAVN;* i teksten.

2.2 KALL-NIVÅER

Det er viktig å forstå begrepet *nivå* i makro-kall og makro-definisjoner.

Et kall startes med et direktiv-start-tegn og avsluttes med et direktiv-slutt-tegn *på samme nivå*.

Eks-3: *ˆMD/NAVN/Pettersen;*

Navn : *NAVN*
Stamme : *Pettersen*
Kall : *ˆNAVN;*
Resultat: *Pettersen*

Definisjonen til denne makroen har bare ett nivå, der *elementene* i definisjonen er *NAVN* og *Pettersen*. Dvs. at det vil bli definert en bruker-makro med navnet *NAVN* og stamme *Pettersen*.

I makro-kall er det ofte hensiktsmessig å ha med kall til andre makroer. Det er f.eks. tilfelle i eksemplet under, der resultatet av kall til makroen *ˆNAVN* er avhengig av makroen *ˆETTERNAVN*.

Eks-4: *ˆMD/ETTERNAVN/Olsen;*
ˆMD/NAVN/Hans navn var ˆETTERNAVN;;

Navn : *NAVN*
Stamme : *Hans navn var Olsen*
Kall : *ˆNAVN;*
Resultat: *Hans navn var Olsen*

Når NOTIS-TF leser definisjonen av makroen, vil systemet treffe på et nytt direktiv-start-tegn før direktiv-slutt-tegnet, og et kall til *ˆETTERNAVN;* som ligger *ett nivå høyere* vil bli startet.

Definisjonen over inneholder altså to kall som ligger på forskjellig nivå, og NOTIS-TF vil behandle de to kallene på følgende måte:

Først vil definisjonen ˆMD påbegynnes, men når systemet kommer til det neste direktiv-start-tegnet, vil det nye kallet *ˆETTERNAVN;* påbegynnes.

Lesingen av ˆMD blir da "lagt til side", og *ˆETTERNAVN;* vil bli behandlet. Siden dette kallet ikke ligger på laveste nivå (1), vil den teksten som blir resultatet av *ˆETTERNAVN;* bli skrevet inn som en del av stammen i kallet *ˆNAVN;*. Når kallet *ˆETTERNAVN;* er avsluttet vil kallet på det laveste nivået (definisjonen, ˆMD) bli "hentet fram" igjen og fullført.

I eksemplet over inneholder stammen i definisjonen et nytt kall til en ny makro. Vi kan tenke oss at denne makroen igjen kan inneholde kall til nok en makro, den igjen til nok en makro, osv. På denne måten kan du neste kall i et "uendelig" antall nivåer (men det gjelder å holde tunga rett i munnen!).

Som du kan se av eksemplet over, vil kallet på det øverste nivået *alltid* bli ferdig behandlet først. Man kan sammenlikne en makro-ekspansjon med et aritmetisk uttrykk: det er alltid den innerste parentesen som blir utregnet først.

En makro-definisjon må alltid være balansert, dvs. at den må inneholde det samme antall start-direktiv-tegn og slutt-direktiv-tegn, og det samme antall start-sitattegn og slutt-sitattegn (se neste seksjon).

2.3 SITATTEGN ("DIRECTIVE QUOTES")

I NOTIS-TF betegnes sekvensene '^<' og '^>' som henholdsvis *start-sitattegn* og *slutt-sitattegn*. Kort forklart brukes disse sitattegnene til å hindre ekspandering av en kall-sekvens. I mange tilfeller kan dette være ønskelig, bl.a. når du definerer makroer som krever parametre.

La oss se på følgende eksempler:

Eks-5: *^MD/ETTERNAVN/Olsen;*
^MD/NAVN1/^ETTERNAVN;;

Navn: NAVN1
Stamme: Olsen
Kall: ^NAVN1;
Resultat: Olsen

Dette eksemplet er likt det vi hadde i det forrige eksempelet, eksempel 4. Vi har en makro-definisjon med et nytt makro-kall i stammen. Definisjonen blir behandlet på følgende måte:

NOTIS-TF leser definisjonen fram til nytt-direktiv-tegnet i makro-kallet *^ETTERNAVN;*. Da avbrytes det første kallet, og *^ETTERNAVN;* behandles. Resultatet blir så satt inn i stammen til definisjonen, og makroen *NAVN1* er nå definert. Den vil ha som resultat *Olsen* hver gang den kalles.

La oss nå se på hva som skjer hvis du bruker sitattegn rundt makro-kallet i stammen:

Eks-6: *^MD/ETTERNAVN/Olsen;*
^MD/NAVN2/^<^ETTERNAVN;^>;

Navn: NAVN2
Stamme: ^ETTERNAVN;
Kall: ^NAVN2;
Resultat: Olsen

Som du ser blir det endelige resultatet det samme som i forrige makro-kall. Men de to blir likevel behandlet på forskjellig måte, og dette er det viktig å forstå. Denne makro-definisjonen blir behandlet på følgende måte:

> NOTIS-TF leser fram til første start-sitattegn påtreffes. Da vil alle etterfølgende tegn fram til tilhørende slutt-sitattegn *bli betraktet som ren tekst*. Dvs. at eventuelle kall som befinner seg mellom sitattegnene vil bli betraktet som ren tekst, og vil følgelig ikke bli ekspandert i første omgang. Dette gjelder også sitattegn inne i sitattegn.
>
> Stammen i makroen blir altså *ˆETTERNAVN;*. Hver gang makroen *ˆNAVN2; kalles* vil *ˆETTERNAVN;* bli ekspandert. Resultatet av *ˆNAVN2;* vil til enhver tid avhenge av definisjonen av *ˆETTERNAVN;*.

La oss illustrere forskjellen mellom de to typene med nok et eksempel:

Eks-7: *Vi har altså nå en situasjon der kallet* **ˆNAVN1;** *gir som* **Olsen** *som resultat, og* **ˆNAVN2;** *gir også resultatet* **Olsen**.

La oss se på hva som skjer hvis vi forandrer definisjonen for **ˆETTERNAVN;**:

ˆMD/ETTERNAVN/Jensen;

Kallet ˆNAVN1; gir fremdeles Olsen som resultat. Kallet ˆNAVN2; gir nå Jensen som resultat.

Forskjellen mellom de to metodene er altså at skrevet **uten** *sitattegn blir resultatet av et makro-kall fast definert ved definisjonsøyeblikket, mens skrevet* **med** *sitattegn er resultatet avhengig av den til enhver tid gyldige definisjon av ˆETTERNAVN;*

Det er viktig å merke seg at selve sitattegnene forsvinner etter at makroen er ekspandert.

I eksemplet over betyr det at stammen til makroen *NAVN2* blir *ˆETTER-NAVN;* og ikke *ˆ<ˆETTER-NAVN;ˆ>*.

Ved bruk av sitattegn i makro-definisjoner trenger sitattegnene ikke nødvendigvis å omslutte hele stammen. Dette kan illustreres med følgende eksempel:

Eks-8: *ˆMD/ETTERNAVN/Olsen;*
ˆMD/FORNAVN/Rune;
ˆMD/MELLOMNAVN/Th. ;
ˆMD/NAVN2/ˆ<ˆFORNAVN;ˆ> ˆMELLOMNAVN;ˆ<ˆETTERNAVN;ˆ>;

Navn: NAVN2
Stamme: ˆFORNAVN; Th. ˆETTERNAVN;
Kall: ˆNAVN2;
Resultat: Rune Th. Olsen

I dette tilfellet vil ekspanderingen av *FORNAVN* og *ETTERNAVN* bli forsinket fordi disse er skrevet mellom sitattegn, mens *MELLOMNAVN* vil bli ekspandert ved definisjonen av *NAVN2*. Den samme teknikken kan benyttes dersom man ønsker å inkludere verdier av system-makroer eller integer-makroer ved definisjonstidspunktet.

La oss så se på et eksempel med en makro som krever en parameter når den kalles:

Eks-9: *ˆMD/TITTEL/ˆ<Manual for ˆ1;ˆ>;*

Navn : TITTEL
Stamme : Manual for ˆ1;
Kall : ˆTITTEL/NOTIS-TF;
Resultat: Manual for NOTIS-TF

Kallet ˆ1; er et spesialkall som betyr at her skal det settes inn en parameter. Som du ser her var det viktig at kallet ˆ1; ikke ble utført ved definisjons-øyeblikket, men ble liggende i stammen. ˆ1; er enda ikke definert ved definisjonstidspunktet. Resultatet av makroen avhenger altså av parameteren som gis når den kalles. Mer om parameterinkludering på side 24.

På side 23 og side 27 finner du eksempler på praktisk anvendelse av sitattegn, og som du vil se der er det viktig å kunne bruke sitattegn for å bygge opp litt avanserte makroer.

Hver gang et start-sitattegn påtreffes vil behandlingen fortsette ett nivå høyere. Behandlingen av dette nivået avsluttes ikke før tilhørende slutt-sitattegn påtreffes. Eventuelle start-sitattegn inni sitattegn fører til at behandlingen går enda et nivå opp.

Makro-definisjoner må være balanserte. Dvs. at det må være like mange start-sitattegn som slutt-sitattegn.

3 INNGÅENDE BESKRIVELSE AV MAKRO-TYPEN SYSTEM-MAKRO

Dette er den enkleste makro-typen, på den måten at du ikke kan definere eller forandre den, bare kalle den.

En system-makro kalles på følgende måte:

`*$Navn/Type;*

Her følger en liste over alle system-makroer med en kort beskrivelse av deres funksjon:

KALL	BESKRIVELSE	FORMAT
$PN	Aktuelt sidenummer	numerisk verdi
$CN	Aktuelt kapittelnummer	numerisk verdi
$SN	Aktuelt seksjonsnummer	numerisk streng
$AN	Aktuelt vedleggsnummer	numerisk verdi
$CPOS	Aktuell linjeposisjon	numerisk verdi
$CLINE	Aktuelt linjenummer	numerisk verdi
$SECLEV	Aktuelt seksjonsnivå	numerisk verdi
$LM	Venstre marg	antall tegn
$RM	Høyre marg	antall tegn
$LB	Venstre kant	antall tegn
$OB	Høyre kant	antall tegn
$TB	Øvre kant	antall linjer
$BB	Nedre kant	antall linjer
$PL	Sidelengde	antall linjer
$PW	Sidebredde	antall tegn
$TW	Tekstbredde = $PW - $LB - $OB	antall tegn
$SI	Seksjons-innrykk	antall tegn
$SS	Seksjons-avstand	antall linjer
$SF	Minimum seksjons-størrelse	antall linjer
$PI	Avsnitts-innrykk	antall tegn
$PS	Avsnitts-avstand	antall linjer
$PF	Minimum avsnitts-størrelse	antall linjer
$BT	Nivå for uthevet skrift	numerisk verdi
$BS	Nivå for uthevet seksjonshode	numerisk verdi
$LS	Linje-avstand	antall linjer
$HP	Tegntetthet	tegn/tomme
$VP	Linjetetthet	linjer/tomme
$TI	Tittel	tekst
$AU	Forfatter	tekst
$DI	Distribueringsliste	tekst
$TO	Mottaker	tekst
$LH	Brevhode	tekst
$RF	Referanse	tekst
$AS	Omtale	tekst
$CH	Kapittel hode	tekst
$H1	Sidehode linje 1	tekst
$H2	Sidehode linje 2	tekst
$TL	Sidehale	tekst

$CHEAD	Innholdsfortegnelsehode	tekst
$DX	Flagg for dobbeltsidig tekst	aktiv hvis > 0
$F-PH-STR	Flagg for PH-STR,	aktiv hvis > 0
$F-TL-STR	Flagg for TL-STR	aktiv hvis > 0
$F-EOD-STR	Flagg for EOD-STR	aktiv hvis > 0
$F-EOF-STR	Flagg for EOF-STR	aktiv hvis > 0
$F-PAR-STR	Flagg for PAR-STR	aktiv hvis > 0
$F-UND-STR	Flagg for UND-STR	aktiv hvis > 0
$DATE	Dato	20.07.1983
$YEAR	Fullt år	1983
$YR	Delvis år	83
$MM	Måned	07
$M	Enkel måned	7
$DD	Dag	02
$D	Enkel dag	2
$FDATE	Fil dato	20.07.1983
$FYEAR	Fil fullt år	1983
$FYR	Fil delvis år	83
$FMM	Fil måned	07
$FSM	Fil enkel måned	7
$FDD	Fil dag	02
$FSD	Fil enkel dag	2
$TIME	Tid (engelsk)	12:45 am.
$TID	Tid (norsk)	00.45
$HOUR	Time	00
$MIN	Minutt	45
$SEC	Sekund	36

Alle disse system-makroene kan ha en parameter til kallet. Denne parameteren kan ha en av følgende verdier:

A : Alfabetisk representasjon med store bokstaver.

a : Alfabetisk representasjon med små bokstaver.

R : Representert i romertall med store bokstaver.

r : Representert i romertall med små bokstaver.

N : Numerisk representasjon.

n : Numerisk representasjon.

Her er numerisk verdi standardverdien. Representasjon i romertall har en begrensning oppad til 3999. Alfabetisk representasjon er begrenset oppad til 18278.

Alfabetisk representasjon virker på følgende måte:

1=a, 2=b, 3=c,.....26=z
27=aa, 28=ab, 29=ac.....52=az
53=ba, 54=bb, 55=bc.. osv.

Eks-10: *Hvis du i teksten din skriver:*

Dette er side ˆ$PN;, *vil resultatet bli:*

Dette er side 19, *der* **19** *er nummeret på den aktuelle siden i det formaterte dokumentet.*

Hvis du i teksten din skriver:

Dette er side ˆ$PN,a;, *vil resultatet bli:*

Dette er side s, *der* **s** *er nummeret på den aktuelle siden i det formaterte dokumentet.*

Hvis du i teksten din skriver:

Dette er side ˆ$PN,R;, *vil resultatet bli:*

Dette er side XIX, *der* **XIX** *er nummeret på den aktuelle siden i det formaterte dokumentet.*

System-makroen $SN er derimot et spesialtilfelle. Denne makroen kan ha to parametre til kallet, der den første parameteren angir skilletegnet mellom numrene på forskjellig seksjonsnivåer.

Eks-11: *Hvis du skriver:*

I denne seksjonen, ˆ$SN,/;,...

blir resultatet, for eksempel:

I denne seksjonen, seksjon 3/1/2,....

Kallet **ˆ$SN=)=a;** *gir resultatet* **c)a)b**.

Parameteren for tallsystem blir altså parameter to til denne makroen.

Kallet **ˆ$SN;**, *uten parametre, gir standard-resultatet* 3.1.2.

4 INNGÅENDE BESKRIVELSE AV BRUKER-MAKROER

Det finnes en del direktiver som er direkte knyttet til behandling av bruker-makroer:

MD : Makro Define, definerer en brukermakro.

PM : Parameter Include, henter en parameter til kallet.

MK : Makro Kill, sletter en brukermakro.

MA : Makro Append, utvider definisjonen av en brukermakro.

MI : Makro Insert, utvider definisjonen av en brukermakro.

MR : Makro Remove, fjerner en del av en brukermakro-definisjon.

CM : Clear Makro, fjerner hele brukermakro-stammen.

DM : Dump Makro, skriver makro-definisjonen på skjermen.

FIX : "Fikserer" (låser) en bruker-makro-definisjon, dvs at den ikke kan slettes.

NREDEF: No-Redefine, hindrer redefinering av bruker-makroen.

4.1 MD, MAKRO DEFINE

Dette direktivet brukes til å definere en bruker-makro, og er et viktig og mye brukt direktiv.

En bruker-makro virker slik:

1) Først må makroen *defineres*. Definisjonen må enten skrives inn i tekstdokumentet, eller inkluderes i en modifisert utgave av det standard makro-biblioteket (i filen NOTIS-TF-NOR-xx:LIB) som er en del av systemet.

2) *Kallet* skrives så inn i teksten på det ønskede stedet, og makroen utføres (ekspanderes) når dokumentet formateres. I det formaterte dokumentet vil da kallet være byttet ut med den teksten som blir resultatet av makroen.

Sagt på en annen, litt mer teknisk måte: man kan tenke seg at det tekstdokumentet som leses av NOTIS-TF blir "lagt til side" når en makro påtreffes, og makroens stamme blir lest. Når så makro-stammen er lest og utført, hentes tekstdokumentet fram igjen, og lesingen fortsetter der den ble avbrutt, altså etter makro- kallet. Man kan derfor sammenlikne et bruker-makro-kall med inkludering av et nytt dokument, der makro-stammen er det nye dokumentet.

Bruker-makroen består av to deler: *Navn* og *Stamme*, og defineres på følgende måte:

^MD/Navn/Stamme;

Navn kan bestå av en kombinasjon av bokstaver, tall og tegnet '-' (bindestrek). Men et navn må alltid begynne med en bokstav, og slutte med en bokstav eller et tall.

Stammen kan bestå av en kombinasjon av tekst og direktiver/makroer.

4.1.1 BRUKERMAKRO MED KONSTANT STAMME

Den enkleste formen for bruker-makro er en makro med en konstant tekst.

Eks-12: *ˆMD/PRODUKT/NOTIS-TF;*

Navn : *PRODUKT*
Stamme : *NOTIS-TF*
Kall : *Dette er en beskrivelse av ˆPRODUKT;*
Resultat: *Dette er en beskrivelse av NOTIS-TF*

Som du ser vil kallet til makroen i tekstdokumentet bli byttet ut med makroens stamme i det formaterte dokumentet.

4.1.2 BRUKERMAKRO MED MAKRO-KALL I STAMMEN

Eks-13: *ˆMD/TITTEL/Manual for ˆPRODUKT;;*

Navn : *TITTEL*
Stamme : *Manual for NOTIS-TF*
Kall : *ˆTITTEL;*
Resultat: *Manual for NOTIS-TF*

Resultatet av denne makroen blir også en konstant tekst i stammen, men stammen har en variabel ved definisjonstidspunktet. Dvs. at siden kallet til *PRODUKT* ligger inne i kallet til *MD*, vil *PRODUKT* bli utført først. Dette fører da til at resultatet av *PRODUKT* skrives inn i stammen til *TITTEL*, slik at denne blir en konstant tekst.

Et makro-kall utføres ikke før kallet i sin helhet er lest, dvs. til og med direktiv-slutt-tegnet på det laveste nivået.

I dette tilfellet vil da utføringen/ekspanderingen av *ˆTITTEL;* forløpe på følgende måte:

- Direktiv-start-tegnet starter kallet.
- Deretter leser NOTIS-TF fram til første ikke-alfabetiske, ikke-numeriske tegn, i dette tilfellet ';'.
- NOTIS-TF vil så gå til definisjonen av *TITTEL* og fortsette lesingen der.

Defineringen av *^TITTEL;* vil forløpe på følgende måte:

- Direktiv-start-tegnet starter kallet (definisjonen).

- Deretter leser NOTIS-TF fram til første ikke-alfabetiske, ikke-numeriske tegn, i dette tilfellet '/'. Det som til nå er lest blir betraktet som kallets navn.

- Så leses det fram til neste skilletegn. I dette tilfellet er det ikke noe nytt skilletegn, og systemet vil da lese fram til direktiv-slutt-tegnet. Men før dette er nådd, påtreffes et nytt kall. Behandlingen av MD-kallet blir da midlertidig "lagt til side", og NOTIS-TF skifter til et nytt nivå.

- Kallet til *PRODUKT* blir så lest i sin helhet og behandlet. Dette fører til at resultatet av *PRODUKT* blir skrevet der hvor kallet sto, dvs. i stammen til MD-kallet. Når PRODUKT-kallet så er avsluttet skiftes det tilbake til forrige nivå, altså til MD-kallet.

- Lesingen fortsetter nå der den ble avbrutt, dvs. etter kallet til *PRODUKT*. Dette betyr altså at når stammen til MD-kallet er ferdig lest, er den blitt forandret til en konstant tekst.

- Stammen avsluttes så med direktiv-slutt-tegnet. Kallet er da lest i sin helhet, og behandlingen av det kan begynne.

- Det som nå skal behandles er altså et kall med navnet *MD*, der første parameter er *TITTEL* og stammen er *Manual for NOTIS-TF*.

- Stammen blir så satt inn i den formaterte teksten der kallet *^TITTEL;* stod i tekstdokumentet.

Et kall leses altså i sin helhet før det behandles. Dette betyr da at alle kall på høyere nivåer utføres før hovedkallet blir behandlet.

4.1.3 BRUKERMAKRO MED VARIABEL DEL I STAMMEN - MAKRO-KALL I SITATTEGN

Som neste trinn kan vi ta for oss en makro med en variabel del i stammen.

Eks-14: *^MD/TITTEL/^<Manual for ^PRODUKT;^>;*

Navn : *TITTEL*
Stamme : *Manual for ^PRODUKT;*
Kall : *^TITTEL;*
Resultat: *Manual for NOTIS-TF*

Sitattegnene fører her til at kallet til *PRODUKT* ikke blir utført ved definisjons-tidspunktet, men blir liggende i stammen til MD-kallet.

Dette kan forklares på følgende måte:

> Når lesingen av stammen til MD-kallet startes, påtreffes et start-sitattegn. Dette betyr at alt som leses inntil tilhørende slutt-sitattegn påtreffes betraktes som ren tekst. Kallet til *PRODUKT* blir derfor ikke utført, og *PRODUKT* trenger heller ikke å være definert på dette tidspunkt.

Når så *TITTEL* kalles, kan man forestille seg at tekstdokumentet som leses "legges til side", og *TITTEL*s stamme blir betraktet som tekstdokument. Stammen er nå *Manual for ^PRODUKT;*. Kallet til *PRODUKT* vil da først bli utført når *TITTEL* kalles. Det blir utført akkurat som om det skulle være skrevet direkte i tekstdokumentet.

Vi sier at makroen har en variabel stamme, da den er avhengig av andre makroer.

Man kan tenke seg at makroen *TITTEL* brukes i flere sammenhenger slik som den er, mens definisjonen av *PRODUKT* varieres. Makroens resultat kan derfor forandres uten at man forandrer definisjonen av den. Her ser vi at bruken av sitattegn er viktig. Uten disse kunne man ikke forandre definisjonen av *PRODUKT*.

4.1.4 BRUKERMAKRO MED PARAMETERINKLUDERING

En annen måte å definere en makro med variabel stamme er å ta i bruk parameter-inkludering. Dette er en nyttig mekanisme som er mye brukt.

Eks-15: *^MD/TITTEL/^<Manual for ^1;^>;*

Navn : *TITTEL*
Stamme : *Manual for ^1;*
Kall : *^TITTEL/NOTIS-TF;*
Resultat: *Manual for NOTIS-TF*

Kallet ^1; er et spesielt direktiv som betyr at her skal du sette inn parameter 1 som du skriver i kallet til makroen. På denne måten kan du ha med et "uendelig" antall parametre i en makro ved å nummerere dem fortløpende. Parametrene trenger ikke å defineres i stigende rekkefølge, og én og samme parameter kan brukes flere steder i en og samme makro. Dvs. at en parameter ikke blir "brukt opp" etter at du har inkludert den en gang.

Resultatet av *TITTEL* er nå ikke avhengig av andre makroer, men er avhengig av den parameteren som gis ved kallet til *TITTEL*.

Husk at sitattegnene her er helt nødvendige for at kallet ^1; ikke skal utføres ved definisjonstidspunktet. Parameteren skal settes inn først når makroen kalles.

4.1.5 PARAMETERINKLUDERING MED STANDARDVERDIER

Når du bruker parameter-inkludering kan du også definere en standardverdi for hver parameter. Det betyr at dersom du ikke spesifiserer parameteren ved kallet, vil standard-verdien brukes. Dette gjøres som i eksemplet under:

Eks-16: *ˆMD/TITTEL/ˆ<Manual for ˆ1,NOTIS-WP;ˆ>;*

Navn : *TITTEL*
Stamme : *Manual for ˆ1,NOTIS-WP;*

Kall-1 : *ˆTITTEL/NOTIS-TF;*
Resultat: *Manual for NOTIS-TF*

I kall-1 ble parameteren *NOTIS-TF* gitt. Denne ble da inkludert som parameter 1 i følge definisjonen.

Kall-2 : *ˆTITTEL;*
Resultat: *Manual for NOTIS-WP*

I kall-2 ble det ikke gitt noen parameter. Standard-verdien *NOTIS-WP* ble derfor brukt.

Kall-3 : *ˆTITTEL/;*
Resultat: *Manual for NOTIS-WP*

I kall-3 ble det gitt det vi kaller en *tom parameter*. Det fører også til at standardverdien blir brukt. Tomme parametre er det nødvendig å bruke i tilfeller der det er mer enn en parameter til et kall, og du vil bruke standardverdiene for noen av parametrene. Tomme parametre gis ved at du setter to deletegn helt inn til hverandre. En parameter som inneholder én eller flere blanke (mellomrom) blir IKKE betraktet som tom. Se neste eksempel.

Som nevnt ovenfor kan du inkludere mer enn én parameter i en makro. Dette kan illustreres med følgende eksempel.

Eks-17: *^MD/TITTEL/^<Manual for ^1,NOTIS-WP; versjon ^2,A;^>;*
Navn : TITTEL
Stamme : Manual for ^1,NOTIS-WP; versjon ^2,A;

Kall-1 : ^TITTEL/NOTIS-TF/I;
Resultat: Manual for NOTIS-TF versjon I

I kall-1 ble det gitt parametrene *NOTIS-TF* og *I*. Disse ble da inkludert i følge definisjonen.

Kall-2 : ^TITTEL/NOTIS-TF;
Resultat: Manual for NOTIS-TF versjon A

I kall-2 ble parameter 2 utelatt. Dette førte da til at standard-verdien for denne ble brukt.

Kall-3 : ^TITTEL;
Resultat: Manual for NOTIS-WP versjon A

I kall-3 ble begge parametrene utelatt. Dette førte da til at begge standard-verdiene ble brukt.

Kall-4 : ^TITTEL//;
Resultat: Manual for NOTIS-WP versjon A

I kall-4 ble det gitt 2 tomme parametre. Dette førte til samme resultat som i kall-3.

Kall-5 : ^TITTEL//////////////;
Resultat: Manual for NOTIS-WP versjon A

I kall-5 ble det gitt 14 tomme parametre. Dette førte også til samme resultat som i kall-3. Dette illustrerer at dersom du gir flere parametre enn det som er definert, vil de overskytende parametrene bare bli ignorert.

Kall-6 : ^TITTEL//G;
Resultat: Manual for NOTIS-WP versjon G

I kall-6 ble det gitt en tom parameter 1 mens *G* ble gitt som parameter 2. Dette førte da til at standard-verdien ble benyttet for parameter 1, mens parameter 2 ble inkludert i følge definisjonen.

4.1.6 PARAMETERINKLUDERING VED HJELP AV DIREKTIVET PM

Det finnes også et annet direktiv for inkludering av parametre: *PM*.

Dette direktivet kan ha to parametre: *Nummer* og *standard-verdi*:

ˆPM/nummer/standardverdi;

Nummer angir parameternummeret, og skal være en positiv tallverdi.

For *standardverdi* gjelder samme regler som for en bruker-makro-stamme.

Med følgende eksempel vil vi forsøke å illustrere forskjellen mellom disse to måtene å inkludere parametre.

Eks-18: *ˆMD/TITTEL/ˆ<Manual for ˆ1,NOTIS-WP; versjon ˆPM,2,A;ˆ>;*
ˆMD/PRODUKT/NOTIS-TF;
ˆMD/VERSJON/J;

Navn : *TITTEL*
Stamme : *Manual for ˆ1,NOTIS-WP; versjon ˆPM,2,A;*

Kall-1 : *ˆTITTEL/NOTIS-TF/I;*
Resultat: *Manual for NOTIS-TF versjon I*

Kall-2 : *ˆTITTEL/ˆPRODUKT;/ˆVERSJON;;*
Resultat: *Manual for NOTIS-TF versjon J*

Kall-3 : *ˆTITTEL/ˆ<ˆPRODUKT;ˆ>/ˆ<ˆVERSJON;ˆ>;*
Resultat: *Manual for NOTIS-TF versjon ˆVERSJON;*

I kall-1 ble det gitt en konstant tekst i begge parametrene, og de ble derfor inkludert likt i begge tilfeller.

I kall-2 ble det gitt to makro-kall som parametre. Men disse var ikke skrevet i sitattegn, og ble derfor utført under lesing av kallet. Kallet ble derfor *utført* med to konstante tekster som parametre.

I kall-3 ble det gitt to makro-kall skrevet i sitattegn som parametre. Som du ser ble parametrene i dette tilfellet forskjellig behandlet. Det er i slike tilfeller forskjellen mellom de to måter å inkludere parametre på kommer frem. I dette tilfellet var resultatet av parameter 2 ikke noe ønskelig resultat. Men vi valgte et enkelt eksempel kun for å forklare forskjellen på de to måtene å inkludere parametre.

Det som skiller de to metodene er:

Parameter-inkludering etter den første metoden med *^nummer/standard*; utføres ved at parameteren ekspanderes som et kall. Dvs. at parameteren leses som en stamme i en makro-ekspansjon. Dette fører til at eventuelle kall som befinner sog i parameteren vil bli utført. Som du så i kall-3, ble kallet til makroen *PRODUKT* i parameter 1 utført.

Parameter-inkludering etter den andre metoden, *^PM/nummer/standard;*, *utføres ved at parameteren kopieres inn uten å bli analysert*. Dette fører derfor til at eventuelle kall i parameteren ikke vil bli utført. Som du så i kall-3, ble kallet til *VERSJON* i parameter 2 ikke utført. Vi kan altså si at PM-direktivet har samme virkning som et ekstra sett med sitattegn. Behandlingen av makro-definisjonen vil gå *ett nivå opp* når systemet treffer på et ^PM-direktiv.

Med det følgende eksempelet vil vi illustrere et mer praktisk tilfelle der det kan være ønskelig å inkludere en parameter uten ekspansjon:

Eks-19: *^MD/MAK-DEF/^<^MD,^1;,^PM=2;;^>;*

Navn : *MAK-DEF*
Stamme : *^MD,^1;,^PM=2;;*

Kall-1 : *^MAK-DEF/INNRYKK/^<^BL=2;^LM=+3;^>;*
Resultat: *Det vil bli definert en makro med navn* **INNRYKK** *og stamme* **^BL=2;^LM=+3;**.

Kall-2 : *^MAK-DEF/EKS/^<^NF;^LM=+7;^IP=Eks: ;^>;*
Resultat: *Det vil bli definert en makro med navn* **EKS** *og stamme* **^NF;^LM=+7;^IP=Eks: ;**.

Bruk av PM-direktivet gjør det her mulig å inkludere direktiver som en del av en parameter.

I kapittel 10 på side 78 under eksemplet som omhandler makroen *FIG-CP* finner du et annet eksempel på bruk av PM-direktivet.

4.2 MK, MAKRO KILL

Dette direktivet brukes til å ugyldiggjøre, slette, en makro. Dette betyr at makroen forsvinner fra settet av definerte makroer. Et forsøk på å kalle en *slettet* makro er det samme som å forsøke å kalle en makro som ikke er definert; det vil resultere i en feilmelding.

MK-direktivet krever en parameter, og kalles på følgende måte:

ˆMK/Navn;

der *Navn* er navnet på den makroen som skal slettes.

Når man definerer en makro, vil denne makroen oppta plass i maskinens hukommelse. Hvor mye plass som opptas er avhengig av makroens størrelse. Når man så sletter en makro, frigjøres den plassen den opptok, slik at du får plass til nye makro-definisjoner. Så dersom du jobber med store og mange makroer kan det være nødvendig å rydde opp innimellom ved å slette makroer som ikke skal brukes mer.

Dersom du definerer en makro to (eller flere) ganger, dvs. har to makro-definisjoner med samme navn, vil den siste definisjonen være den som gjelder. Men den første definisjonen vil ikke forsvinne, den vil bare ikke være tilgjengelig. Hvis du så sletter makroen med dette navnet, vil du slette den *siste* definisjonen, og den forrige definisjonen vil igjen bli gjort gyldig. Dette kan illustreres med følgende eksempel:

Eks-20: *ˆMD/NAVN/Olsen;*

Kallet ˆ**NAVN**; *vil nå gi resultatet* **Olsen**

ˆMD/NAVN/Pedersen;

Kallet ˆ**NAVN**; *vil nå gi resultatet* **Pedersen**

ˆMK/NAVN;

Kallet ˆ**NAVN**; *vil nå gi resultatet* **Olsen**

Dersom du forsøker å slette en makro som ikke er definert, vil dette resultere i en feilmelding.

4.3 MA, MAKRO APPEND

Dette direktivet brukes til å utvide stammen i en bruker- eller referanse-makro.

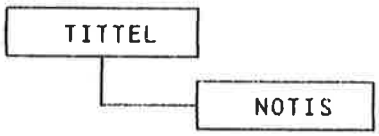
MA-direktivet krever minimum to parametre, og kalles på følgende måte:

ˆMA/Navn/stamme-element/../stamme-element;

der *Navn* er navnet på den makroen som skal utvides, og *stamme-element* er tillegget til den eksisterende makro-stammen. De samme syntaksregler gjelder selvfølgelig her som ved definisjon av en makro.

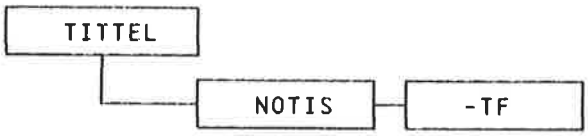
Under vil vi skissere hvordan en bruker- og referanse-makro logisk er oppbygd, slik at det kommer fram hvordan makroen kan utvides.

Definisjonen *ˆMD/TITTEL/NOTIS;* fører til at det blir definert en makro som er bygd opp på følgende måte:



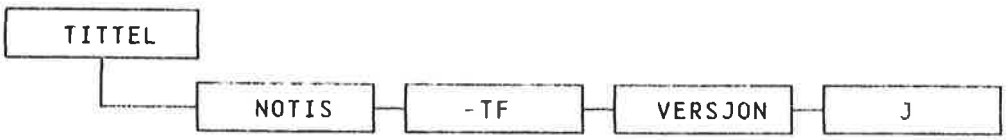
Vi sier da at denne makroen består av to *elementer*, der navnet er ett element og stammen ett.

Kallet *ˆMA/TITTEL/-TF;* vil da i dette tilfellet føre til at makroen utvides på følgende måte:



Som det fremgår av syntaksdefinisjonen for dette kallet, er det mulig å utvide en makro med mer enn ett stamme-element i ett og samme kall.

Kallet *ˆMA/TITTEL/ VERSJON/ J;* vil i dette tilfellet føre til at makroen utvides på følgende måte:



4.4 MI, MAKRO INSERT

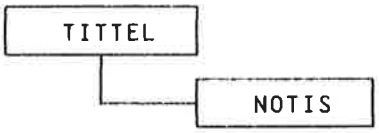
Dette direktivet brukes også til å utvide en stamme i en bruker- eller referanse-makro (se også MA-direktivet, side 30).

MI-direktivet krever minimum to parametre, og kalles på følgende måte:

^MI/Navn/stamme-element/../stamme-element;

der *Navn* er navnet på den makroen som skal utvides, og *stamme-element* er det som skal legges inn i den eksisterende makro-stammen. De samme syntaksregler gjelder selvfølgelig her som ved definisjon av en makro.

Definisjonen *^MD/TITTEL/NOTIS;* fører til at det blir definert en makro som er bygd opp på følgende måte:

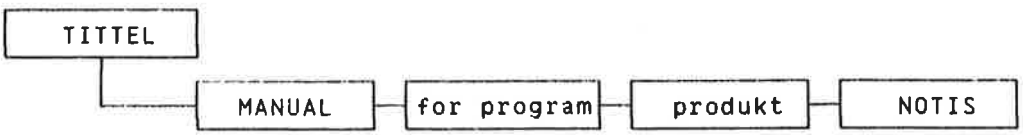


Kallet *^MI/TITTEL/produkt;* vil da i dette tilfellet føre til at makroen utvides på følgende måte:

I prinsippet virker altså dette direktivet på samme måte som MA-direktivet, med den forskjell at de nye stamme-elementene blir lagt inn foran de eksisterende elementene.

Som det framgår av syntaksdefinisjonen for dette kallet, er det mulig å utvide en makro med mer enn ett stamme-element i ett og samme kall.

Kallet *^MI/TITTEL/ MANUAL/ for program;* vil i dette tilfellet føre til at makroen utvides på følgende måte:



4.5 MR, MAKRO REMOVE

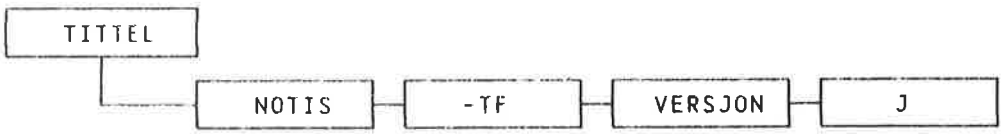
Dette direktivet brukes til å fjerne en del (et element) av stammen i en bruker- eller referanse-makro.

MR direktivet kalles på følgende måte:

^MR/Navn/retning;

der *Navn* er navnet på den makroen som skal reduseres. Parameteren *retning*, som er valgfri, spesifiserer hvordan makro-stammen skal reduseres. Den kan gis en av verdiene F eller B, der F angir at det fremste stamme-elementet skal fjernes, og B, som er standardverdien, angir at det bakerste stamme-elementet skal fjernes.

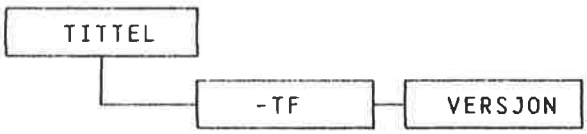
For å illustrere dette kan vi ta utgangspunkt i en makro som ser ut som følger:



Kallet *^MR/TITTEL;* vil da føre til at makroen reduseres til:

I dette kallet ble det ikke spesifisert hvordan makroen skulle reduseres, så standardverdien *B* (bakfra) ble brukt. Dette førte da til at det bakerste elementet i makro-stammen ble fjernet.

Videre vil kallet *^MR/TITTEL/F;* føre til at makroen blir ytterligere redusert på følgende måte:



Som vi ser ble nå det fremste stamme-elementet fjernet.

4.6 CM. CLEAR MAKRO

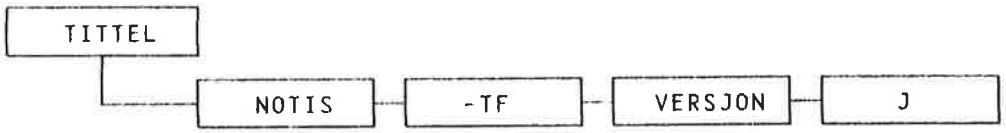
Dette direktivet brukes til å fjerne hele stammen i en bruker- eller referanse-makro.

CM-direktivet kalles på følgende måte:

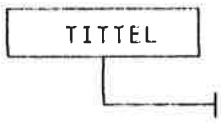
^CM/Navn;

der *Navn* er navnet på den makroen som stammen skal fjernes fra.

For å illustrere dette kan vi ta utgangspunkt i en makro som ser ut som følger:



Kallet *^CM/TITTEL;* vil da føre til at makroen blir seende ut som følger:



Man kan så bruke MA-direktivet for å bygge opp ny stamme til denne makroen.

Dersom man ønsker å redefinere en makro, vil sekvensen CM, MA ofte være raskere og bruke mindre ressurser enn sekvensen MK, MD. Særlig gjelder dette litt større og komplekse makroer. Det er selvfølgelig ikke spesielt merkbart hvis det er en engangsforeteelse. Men dersom dette skal gjøres mange ganger, f.eks. inni makro som kalles ofte, kan man på denne måten redusere ressursforbruket i NOTIS-TF noe.

4.7 DM, DUMP MAKRO

Dette direktivet kan være et godt hjelpemiddel dersom du skulle miste oversikten og få problemer med makro-håndteringen. Ved å bruke dette direktivet kan du få skrevet ut alle viktige data om en makro på et hvilket som helst tidspunkt under formateringen. Data vil bli skrevet ut på skjermen, og formateringen vil stoppe og vente på beskjed fra deg om den skal fortsette eller avbryte.

DM-direktivet kalles på følgende måte:

^DM/Navn;

der *Navn* er navnet på den makroen som skal dumpes ut på terminalen.

4.8 FIX, FIKSÉR MAKRO

Dette direktivet "fiksérer" eller "fryser" en makro-definisjon. Dette betyr at denne versjonen av makroen ikke kan slettes. Dvs. at dersom du definerer en ny makro med samme navn, vil fikseringen ikke gjelde denne. En fiksering kan ikke omgjøres.

FIX-direktivet kalles på følgende måte:

'FIX/navn/navn/.../navn;

der *navn* er navnet på en makro som skal fryses. Som det fremgår av syntaksbeskrivelsen, er det mulig å fryse flere makroer i ett kall.

4.9 NREDEF, SET NO-REDEFINE

Med dette direktivet kan du bestemme at en makroen ikke kan redefineres.

NREDEF-direktivet kalles på følgende måte:

`*NREDEF/navn/navn/.../navn;*

der *navn* er navnet på en makro som ikke skal kunne redefineres. Som det fremgår av syntaksbeskrivelsen, er det mulig å sette "no-redefine" på flere makroer i ett kall.

Dette direktivet kan det være nyttig å bruke i et makro-bibliotek for å reservere makronavn hvis definisjoner er skjult for andre brukere.

5 INNGÅENDE BESKRIVELSE AV MAKRO-TYPEN REFERANSE-MAKRO

Det finnes en del direktiver som er direkte knyttet til behandling av referanse-makroer:

RD : Reference Define, definerer en referanse-makro.

MK : Makro Kill, sletter en referanse-makro.

MA : Makro Append, utvider definisjonen av en referanse-makro.

MI : Makro Insert, utvider definisjonen av en referanse-makro.

MR : Makro Remove, fjerner en del av en referanse-makro-definisjon.

CM : Clear Makro, fjerner hele stammen i en referanse-makro.

DM : Dump Makro, skriver makro-definisjonen på skjermen.

FIX : Fikserer en referanse-makro-definisjon, kan ikke slettes.

NREDEF: No-Redefine, hindrer redefinering av referanse-makroen.

5.1 RD, REFERENCE DEFINE

Dette direktivet brukes til å definere en referanse-makro.

Oppbyggingen av en referanse-makro kan direkte sammenliknes med en bruker-makro. Den eneste forskjellen er at referanse-makroen kan kalles før den er definert.

Syntaksen for definisjon av en referanse-makro er derfor som følger:

^RD/Navn/Stamme;

der reglene er de samme som for bruker-makroer.

En referanse-makro kalles på følgende måte:

^#Navn/Størrelse;

der reglene for navn er de samme som for bruker-makroer, og *Størrelse* er det antall posisjoner som trengs til henvisningen.

Eks-21: *La oss som et eksempel se på en referanse-makro for side-nummer-henvisning. Du holder på med introduksjonen til et emne, og vil ha med en sidehenvisning til hvor dette emnet vil bli utdypet senere i dokumentet. Du skriver for eksempel:*

dette er nærmere forklart på side ^#SIDE-REF/2;..

Dette kallet vil reserve et 2-tegn stort felt i teksten der referansen siden vil bli satt inn. **SIDE-REF** *er et navn som du selv finner opp, og må bruke igjen når du senere definerer makroen.*

Når du så lenger ut i dokumentet kommer til denne utdypingen av emnet, definerer du referanse-makroen på følgende måte:

^RD/SIDE-REF/^$PN;;

Her gir du makroen navnet **SIDE-REF**, *som du allerede har bestemt i kallet til makroen, og stammen inneholder et kall til systemmakroen for sidenummer. Makroen blir altså definert til å inneholde sidenummeret for den siden du i øyeblikket befinner deg på, og uansett hvor mye tekst du legger til eller trekker fra i dokumentet på et senere tidspunkt, vil denne sidehenvisningen alltid være korrekt. Dette sidenummeret vil så bli høyrejustert i det 2-tegns store feltet som du reserverte i kallet til makroen.*

Dersom det viser seg at sidenummeret f.eks. krever 3 tegn, vil linjen med henvisningen bli utvidet med ett tegn, og det vil bli gitt melding om dette under formateringen.

Da oppbyggingen av referanse-makroer er så lik bruker-makroer, vil vi ikke forklare disse nærmere, men henviser til kapitlet for bruker-makroer (se side 20).

5.2 MK. MAKRO KILL

Dette direktivet kan brukes på samme måte for referanse-makroer som for bruker-makroer. (Se side 29.)

5.3 MA. MAKRO APPEND

Dette direktivet kan brukes på samme måte for referanse-makroer som for bruker-makroer. (Se side 30.)

5.4 MI, MAKRO INSERT

Dette direktivet kan brukes på samme måte for referanse-makroer som for bruker-makroer. (Se side 31.)

5.5 MR, MAKRO REMOVE

Dette direktivet kan brukes på samme måte for referanse-makroer som for bruker-makroer. (Se side 32.)

5.6 CM, CLEAR MAKRO

Dette direktivet kan brukes på samme måte for referanse-makroer som for bruker-makroer. (Se side 33.)

5.7 DM, DUMP MAKRO

Dette direktivet kan brukes på samme måte for referanse-makroer som for bruker-makroer. (Se side 34.)

5.8 FIX, FIKSER MAKRO

Dette direktivet kan brukes på samme måte for referanse-makroer som for bruker-makroer. (Se side 35.)

5.9 NREDEF, SET NO-REDEFINE

Dette direktivet kan brukes på samme måte for referanse-makroer som for bruker-makroer. (Se side 36.)

6 INNGÅENDE BESKRIVELSE AV MAKRO-TYPEN INTEGER-MAKRO

Det finnes en del direktiver som er direkte knyttet til behandling av integer-makroer:

IM : Integermakro Define, definerer en integer-makro.

MK : Makro Kill, sletter en integer-makro.

DM : Dump Makro, skriver makro-definisjonen på skjermen.

FIX : Fiksérer en integer-makro-definisjon, kan ikke slettes.

NREDEF: No-Redefine, hindrer redefinering av integer-makroen.

6.1 IM, INTEGERMAKRO DEFINE

Dette direktivet brukes til å definere en integer-makro.

Integer-makroen består av to deler: *Navn* og *Verdi*, og defineres på følgende måte:

'IM/Navn/Verdi;

For *Navn* gjelder samme regler som for makro-navn generelt.

Parameteren *Verdi* kan utelates, i så fall blir verdien satt udefinert. Forøvrig skal denne parameteren bestå av enten en tallverdi alene eller et aritmetisk uttrykk bestående av tallverdier i numerisk representasjon. Romertall og alfabetisk representasjon kan altså ikke brukes i aritmetiske uttrykk.

Reglene for aritmetiske uttrykk er de samme som for AR-direktivet i NOTIS-TF. Se seksjon 9.2 på side 54.

Følgende aritmetiske operatorer kan brukes: ( ) + - * /.

Andre regler: Utregningene foretas fra venstre mot høyre, med parenteser. Antall parentesnivåer er i praksis ubegrenset.

Den enkleste form for definisjon av en integer-makro blir derfor:

Eks-22: *^IM/ANTALL/100;*

```
Navn     : ANTALL
Verdi    : 100
Kall     : ... hvor antallet er ^$ANTALL; ....
Resultat: ... hvor antallet er 100 ....
```

Under er et eksempel på en definisjon med et aritmetisk uttrykk, der det aritmetiske uttrykket brukes for å telle relativt til integer-makroen i eksemplet over:

Eks-23: *^IM/ANTALL-2/^$ANTALL;+1;*

```
Navn     : ANTALL-2
Verdi    : ^$ANTALL; + 1
Kall     : ^$ANTALL-2;
Resultat: 101
```

I denne veiledningen har vi brukt en kombinasjon av integer-makroer og andre makrotyper for å få til automatisk nummerering av eksemplene. I seksjon 10.3 på side 68 viser vi hvordan vi har gjort dette.

6.2 MK, MAKRO KILL

Dette direktivet kan brukes på samme måte for integer-makroer som for bruker-makroer. (Se side 29.)

6.3 DM, DUMP MAKRO

Dette direktivet kan brukes på samme måte for integer-makroer som for bruker-makroer. (Se side 34.)

6.4 FIX, FIKSÉR MAKRO

Dette direktivet kan brukes på samme måte for integer-makroer som for bruker-makroer. (Se side 35.)

6.5 NREDEF, SET NO REDEFINE

Dette direktivet kan brukes på samme måte for integer-makroer som for bruker-makroer. (Se side 36.)

7 INNGÅENDE BESKRIVELSE AV MAKRO-TYPEN TRIGGER-MAKRO

Trigger-makroen er en makro som ikke skal kalles. Den ekspanderer seg selv ("trigges") hver gang de definerte vilkår er tilfredsstilt.

Det finnes en del direktiver som er direkte knyttet til behandling av trigger-makroer:

TM : Trigger-makro Define, definerer en trigger-makro.

MK : Makro Kill, sletter en trigger-makro.

MA : Makro Append, utvider definisjonen av en trigger-makro.

MI : Makro Insert, utvider definisjonen av en trigger-makro.

MR : Makro Remove, fjerner en del av en trigger-makro-definisjon.

DM : Dump Makro, skriver makro-definisjonen på skjermen.

FIX : Fikserer en trigger-makro-definisjon, kan ikke slettes.

ENABLE : Setter en trigger-makro aktiv.

DISABLE: Setter en trigger-makro passiv.

PRIOR : Definerer prioritet for en trigger (variabel).

7.1 TM, TRIGGER-MAKRO DEFINE

Dette direktivet brukes til å definere en trigger-makro.

Trigger-makroen består av 4 deler: *Navn*, *Vilkår*, *Stamme* og *Prioritet*, og defineres på følgende måte:

^TM/Navn/Vilkår/Stamme/Prioritet;

For *Navn* gjelder samme regler som for makro-navn generelt.

Vilkår er et logisk uttrykk som følger samme syntaksregler som logiske uttrykk i direktivet IF. Dette er forklart i seksjon 9.3 på side 56.

For *stamme* gjelder samme regler som for bruker-makro-stamme. Det er imidlertid viktig å merke seg at ekspansjon av stammen MÅ føre til at en eller flere av parametrene i *Vilkår* forandres. Hvis så ikke er tilfelle vil makroen gå i en evig løkke. Da en triggermakro aldri kalles eksplisitt, er det ikke tillatt å inkludere parametre.

Parameteren *prioritet* bestemmer hvilken prioritet makroen skal ha. Denne parameteren er valgfri, med standardverdi 0, og kan anta en verdi i området 0 til 32768. Ved å sette prioritet på trigger-makroene kan du bestemme den innbyrdes rekkefølgen de skal ekspanderes i dersom det oppstår konflikt. En konflikt kan oppstå dersom to makroer får sine vilkår tilfredstilt samtidig.

Et vilkår i en trigger-makro bygges opp av relasjoner mellom variable og konstanter. Det er da selvfølgelig et krav at vilkåret inneholder minst en variabel (trigger). En trigger kan enten være en integer-makro eller en av følgende system-makroer:

$PN
$CN
$CLINE
$SECLEV

Når en trigger-makro defineres vil systemet lage en oversikt over hvilke triggere (variable) som inngår i uttrykket. Hver gang en av disse triggerne modifiseres, vil så alle trigger-makro vilkår hvor denne inngår bli testet, og for hvert vilkår som tilfredsstilles vil tilhørende makro trigges.

Eks-24: *^TM/MERKE/^<^$CLINE; = 5^>/^<-----^BL;^>;*

trigger-makroens vilkår inneholder her en trigger (variabel), $CLINE. Dette betyr at hver gang denne triggeren modifiseres, dvs. hver gang linjenummeret forandrer seg, vil denne trigger-makroens vilkår bli sjekket. I de tilfeller vilkåret blir tilfredsstilt, vil stammen bli ekspandert. Dette betyr at linje 5 på hver side kun vil inneholde

-----

Når en trigger-makro ekspanderes kan den ikke bli avbrutt av andre trigger-makroer før hele stammen er ekspandert ferdig. Dersom ekspansjon av en trigger-makro fører til at én eller flere triggere blir modifisert, kan det derfor oppstå en situasjon hvor det ligger flere trigger-makroer i kø for å bli ekspandert.

Eks-25: *^TM/MERKE/^<^$CLINE; = 5^>/^<-----^BL,1;^>;*
^TM/LINJE/^<^$CLINE; = 6^>/^<.....^bl;^>;

I dette tilfellet har vi to trigger-makroer som begge to er avhengig av triggeren $CLINE. Makroen MERKE trigger på linje 5 og lager 2 linjer. Dette betyr at den også lager linje 6 før den har ekspandert ferdig. Makroen LINJE kan derfor ikke slippe til der hvor den skal, dvs. på linje 6. Den blir derfor lagt i kø, og må vente til makroen MERKE er ferdig.

Dette er et enkelt eksempel, men lignende situasjoner kan føre til at det ligger mange trigger-makroer i kø for å slippe til.

Rekkefølgen i denne køen er avhengig av de impliserte triggeres prioritet. Denne prioriteten kan man sette ved hjelp av direktivet PRIOR. Se seksjon 7.10. Dersom det oppstår en situasjon der to trigger-makroer er trigget av samme variabel (f.eks. $CLINE), vil rekkefølgen av disse to avhenge av trigger-makroenes prioritet. Denne kan spesifiseres i triggermakro-definisjonen.

Det er i denne sammenheng viktig å merke seg at dersom en trigger-makro skal ekspanderes i en løkke når den først trigges, vil den bli lagt inn i køen mellom hver ekspansjon. Dette betyr altså at en trigger-makro kan avbrytes mellom to ekspansjoner av stammen.

Det er kansje på sin plass å komme med et fullstendig eksempel:

Eks-26:
```
^TM/LINJE/^<^$CLINE; = 10^>/^<^PRINT-LINE;^>;
^TM/SIDE/^<(^$PN; = 5) AND (^$FIGSIZE <= 15)^>/
^<^PRINT-FIG;^>;
^PRIOR/CLINE/10;
^PRIOR/PN/5;
```

Makroen PRINT-LINE kan vi tenke oss genererer en linje med et eller annet resultat.

Makroen PRINT-FIG kan vi tenke oss går 15 ganger i løkke, hvor hver løkke genererer 1 linje.

På alle sider fram til side 5 vil makroen LINJE trigges på linje 10. Når man så kommer til side 5 vil makroen SIDE trigges, og den vil starte å produsere et resultat. Når man så kommer til linje 10 vil makroen LINJE trigges. SIDE ble trigget av $PN, mens LINJE ble trigget av $CLINE. Siden $CLINE har høyest prioritet av disse to, vil makroen LINJE slippe til. Når denne så er ferdig, får makroen SIDE fortsette å gjøre seg ferdig.

Dette var et eksempel på hvordan prioritet på triggere virker. Som tidligere nevnt finnes det også en prioritet på selve trigger-makroen. Vi vil prøve å illustrere dette også med et eksempel:

Eks-27:
```
^P-LINE=10;
^TM/LINJE/^<^$CLINE; = 10^>/^<^PRINT-LINE;^>/10;
^TM/FIG-T/^<^$CLINE; = ^$P-LINE;^>/ ^<^PRINT-FIG;^>/5;
```

I dette tilfellet vil begge trigger-makroene trigges av $CLINE når denne antar verdien 10. Her er altså trigger-prioriteten den samme. Siden LINJE har høyest prioritet, vil denne slippe til først. Når denne er ferdig har CLINE fått verdien 11, men siden FIG-T allerede ligger i køen vil den bli ekspandert allikevel.

Dersom en trigger-makro, som produserer et resultat, forsøker å avbryte et implisitt linjeskift som følge av full tekstlinje, vil dette i de fleste tilfeller få et dårlig utfall. Et implisitt linjeskift oppstår i de tilfeller der NOTIS-TF "pakker" linjene, dvs. ved bruk av betinget eller full fyllingsmodus der brukeren ikke spesifiserer hvor linjeskiftene skal være. Det er derfor i slike tilfeller nødvendig å bruke FM=N. Linjelayouten kan da på forhånd formes i tekst-editoren. Dersom man bruker trigger-makroer der $CLINE inngår, er det derfor et krav at man bruker FM=N.

7.2 MK, MAKRO KILL

Dette direktivet kan brukes på samme måte for trigger-makroer som for bruker-makroer. (Se side 29.)

7.3 MA, MAKRO APPEND

Dette direktivet kan brukes på samme måte for trigger-makroer som for bruker-makroer. (Se side 30.)

7.4 MI, MAKRO INSERT

Dette direktivet kan brukes på samme måte for trigger-makroer som for bruker-makroer. (Se side 31.)

7.5 MR, MAKRO REMOVE

Dette direktivet kan brukes på samme måte for trigger-makroer som for bruker-makroer. (Se side 32.)

7.6 DM, DUMP MAKRO

Dette direktivet kan brukes på samme måte for trigger-makroer som for bruker-makroer. (Se side 34.)

7.7 FIX, FIKSÉR MAKRO

Dette direktivet kan brukes på samme måte for trigger-makroer som for bruker-makroer. (Se side 35.)

7.8 DISABLE, SETTER EN TRIGGER-MAKRO PASSIV

Med dette direktivet kan man passivisere en trigger-makro. Å passivisere en trigger-makro vil si at den ikke vil bli trigget uansett de impliserte triggeres tilstand.

Syntaksen for direktivet er som følger:

^DISABLE/navn/navn/.../navn;

der *navn* er navnet på en trigger-makro som skal passiviseres. Som det framgår av syntaksen er det mulig å passivisere flere trigger-makroer i ett og samme kall.

7.9 ENABLE, SETTER EN TRIGGER-MAKRO AKTIV

Dette direktivet er det motsatte av DISABLE. Dvs. at det setter en passiv trigger-makro aktiv.

Syntaksen for direktivet er som følger:

^ENABLE/navn/navn/.../navn;

der *navn* er navnet på en trigger-makro som skal settes aktiv. Som det framgår av syntaksen er det mulig å sette flere trigger-makroer aktiv i ett og samme kall.

7.10 PRIOR, SETTER PRIORITET PÅ EN TRIGGER (VARIABEL)

Syntaksen for dette direktivet er som følger:

^PRIOR/navn/prioritet;

der *navn* er navnet på en integer- eller system-makro som kan brukes som en trigger i en trigger-makro.

Prioritet er en tallverdi i området 0 - 32767.

7.11 SINGLE-TRIGGERS

Det finnes et spesialtilfelle av trigger-makroer, såkalte single-triggers (ST). Dette er en makro som ekspanderes på samme måte som trigger-makroen, men den sletter seg selv etter første ekspandering. En single-trigger ekspanderes aldri mer enn én gang. Etter ekspandering må den defineres på nytt hvis den skal brukes igjen.

En single-trigger defineres på følgende måte:

^ST/Navn/Vilkår/Stamme/Prioritet;

der parametrene følger samme regler som for trigger-makroer.

8 INNGÅENDE BESKRIVELSE AV MAKRO-TYPEN TRIGGER-STRENGER

Trigger-strenger er en forhåndsdefinert form for single-trigger.

De har forhåndsdefinerte *navn*, *vilkår* og *prioritet*, mens stammen kan defineres av brukeren.

I motsetning til single-triggers blir ikke selve makroen slettet etter at den er ekspandert; det er kun stammen som slettes.

Det finnes en system-makro for hver trigger-streng, et *flagg* som forteller om strengen er i bruk eller ikke. Disse er ment å brukes ved testing med IF-direktivet (se seksjon 9.3). De har verdien 1 dersom trigger-strengen er aktiv, verdien 0 dersom trigger-strengen er passiv.

En trigger-streng-stamme defineres på følgende måte:

^NAVN/stamme;

der *navn* er det forhåndsdefinerte navnet til makroen.

For *stamme* gjelder samme regler som for trigger-makroer.

Følgende trigger-strenger er definert i systemet:

TP-STR : Titlepage string

PH-STR : Pageheader string

TL-STR : Trailer string

EOF-STR: End of file string

EOD-STR: End of document string.

PAR-STR: Paragraph string

UND-STR: Underline string

Det er viktig å merke seg at definisjonen av en trigger-streng-stamme tilsvarer det samme som MA for trigger-makroer. Dvs. at den nye stammen vil bli lagt til etter evt. eksisterende stamme i strengen.

8.1 TP-STR, TITLEPAGE STRING (TITTELSIDE-STRENG)

Dette er en streng som trigges før første tegn blir skrevet i ut-dokumentet. Dvs. at dersom denne trigger-strengen har en definert stamme, vil denne bli ekspandert idet systemet prøver å skrive første tegn i det formaterte dokumentet.

Som det framgår av navnet er denne trigger-strengen ment å brukes til å lage en tittelside for dokumentet.

8.2 PH-STR, PAGEHEADER STRING (SIDEHODE-STRENG)

Denne trigger-strengen trigges etter at øvre kant er ferdigbehandlet, dvs. direkte etter sidehodet.

Det finnes en tilhørende system-makro for denne trigger-stringen:

^$F-PH-STR;

Denne gir verdien 0 ved tom stamme, og verdien 1 ved definert stamme.

Et eksempel på bruk kan være et tilfelle der man ønsker å flytte en tekstbit over på neste side uten å avslutte aktuell side. Se FIG-CP-makroen i kapittel 10.

8.3 TL-STR, TRAILER STRING (SIDEHALE-STRENG)

Denne strengen trigges i første linje i nedre kant. Når denne brukes må du passe på at det også er plass til sidehale-strengen i nedre kant. Sidehale-strengen skal skrives i siste linje i nedre kant.

Det finnes en tilhørende system-makro for denne trigger-strengen:

^$F-TL-STR;

Denne gir verdien 0 ved tom stamme, og verdien 1 ved definert stamme.

En typisk bruk av denne trigger-strengen kan være generering av fotnoter. Se makroen FOTNOTE i makro-biblioteket.

8.4 EOF-STR, END OF FILE STRING

Denne trigger-strengen trigges ved slutten av aktuell innfil. Hver innfil har sin egen EOF-STR. Når man definerer stamme for EOF-STR gjelder derfor denne for den aktuelle innfilen på dette tidspunktet.

Det finnes en tilhørende system-makro for denne trigger-strengen:

^$F-EOF-STR;

Denne gir verdien 0 ved tom stamme, og verdien 1 ved definert stamme.

En typisk bruk av denne kan være tømming av figurkø ved slutten av hver innfil, slik som det er gjort i hjelpemakroen *RYDDOPP* som benyttes av makroen *FIG-CP*.

8.5 EOD-STR, END OF DOCUMENT STRING

Denne trigger-strengen trigges ved slutten av dokumentet, etter en evt. EOF-STR definert for hoved-innfil.

Det finnes en tilhørende system-makro for denne trigger-strengen:

^$F-EOD-STR;

Denne gir verdien 0 ved tom stamme, og verdien 1 ved definert stamme.

Et eksempel på bruk av denne trigger-strengen kan være definisjon av en referansemakro som inneholder sidenummer for siste side. Dette kan brukes f.eks. ved skriving av brev der man ønsker å nummerere sidene etter formelen *Dette er side x av n*, der *n* er det totale antall sider.

Forøvrig kan denne trigger-strengen brukes for diverse "opprydding" som skal gjøres ved slutten av dokumentet.

8.6 PAR-STR, PARAGRAPH STRING (AVSNITTS-STRENG)

Dersom denne trigger-strengen er definert, trigges den ved starten av et nytt avsnitt i stedet for den vanlige avsnittshåndteringen. Dvs. at det vil ikke bli foretatt noen avsnittsjustering.

Det finnes en tilhørende system-makro for denne trigger-strengen:

^$F-PAR-STR;

Denne gir verdien 0 ved tom stamme, og verdien 1 ved definert stamme.

8.7 UND-STR, UNDERLINE STRING (UNDERSTREKINGS-STRENG)

Dersom denne trigger-strengen er definert, trigges den ved start/stopp understreking, i stedet for den vanlige understrekings-håndteringen, dvs. at det vil ikke bli foretatt noen understreking av/på.

Det finnes en tilhørende system-makro for denne trigger-strengen:

^$F-UND-STR;

Denne gir verdien 0 ved tom stamme, og verdien 1 ved definert stamme.

Et eksempel på bruk kan være redefinering av understreking til å være skift av skrifttype, slik at understrekede tekstdeler i inn-dokumentet vil komme ut som tekst med en annen skrifttype i ut-dokumentet.

9 SPESIELLE DIREKTIVER SOM ER MYE BRUKT I MAKRO-HÅNDTERING

I dette kapitlet vil vi forklare bruken av en del spesielle direktiver og teknikker som er mye brukt i makro-definering. Vi vil også komme videre inn på nivåer og bruk av sitattegn.

9.1 DIREKTIVENE OF OG NF

Først vil vi forklare funksjonen av direktivene *NF* og *OF*, da disse er mye brukt i forbindelse med makro-håndtering (se også NOTIS-TF Håndbok - Tekstformaterer, seksjon 3.1.14):

Direktivet NF fører til at en del formatparametre blir lagret unna. Disse formatparametrene er:

Venstre marg	($LM)
Høyre marg	($RM)
Fyllings modus	($FM)
Justerings modus	($JM)
Venstre kant	($LB)
Høyre kant	($OB)
Sidebredde	($PW)
Tekstbredde	($TW)
Horisontal tegntetthet	($HP)
Linjetetthet	($VP)
Understrekings status	(Understreking av/på)
Uthevet-skrift status	($BT)
Fontnummer	($FONT)

Etter NF-direktivet kan du så legge inn en del endringer i dette formatsettet. Et kall til direktivet OF fører så til at det forrige formatparametersettet som ble lagret blir hentet frem igjen og brukt som aktuelt sett. Settet med de modifiserte verdiene vil forsvinne.

Det er mulig å neste NF og OF kall. Det må da selvfølgelig ikke forekomme flere OF enn NF kall, det motsatte er derimot ikke så nøye.

9.2 DIREKTIVET AR

AR-direktivet brukes til å utføre aritmetiske beregninger på tall i numerisk representasjon. Det betyr at romertall eller alfabetisk representerte verdier ikke kan brukes i disse beregningene. Direktivet består av to deler, Navn og Uttrykk, og kalles på følgende måte:

^AR/Uttrykk;

Utrykk kan bestå av en kombinasjon av tallverdier og følgende operatorer:

+ Addisjon

- Subtraksjon

/ Divisjon

* Multiplikasjon

() Parenteser for prioriteringsrekkefølge

De generelle regler er at uttrykket utføres fra de innerste parentesnivåene og utover. For operatorer på samme nivå, har multiplikasjon og divisjon samme prioritet, som er høyere enn addisjon og subtraksjon, som har samme innbyrdes prioritet. Dersom det forekommer flere operatorer på samme parentesnivå med samme prioritet, utføres uttrykket fra venstre mot høyre.

Hvis en eller flere høyreparenteser er utelatt i uttrykket, vil disse bli satt inn på slutten slik at beregningen kan bli utført. Resultatet kan da være galt, naturligvis, men du vil bli gjort oppmerksom på dette under formateringen.

Hvis to eller flere etterfølgende operatorer uten argumenter finnes i uttrykk som f.eks. (2+-*3), vil den første bli valgt. Du vil da få melding under formateringen. Operatorer uten argumenter (2+3*) vil bli ignorert og en feilmelding skrevet ut under formatering. Multiplikasjonstegnet foran et uttrykk i parenteser kan utelates.

Eks-28: *^AR/19 + 3(^$LØNN; - 4);*

Det aritmetiske uttrykket blir selvfølgelig ikke beregnet før hele AR-kallet er innlest. I dette eksemplet fører dette til at uttrykket som skal beregnes blir seende ut slik:

19 + 3(n - 4)

der **n** *er verdien av integer-makroen* **LØNN**. *Dette utrykket vil bli beregnet på følgende måte:*

Verdien 4 trekkes fra **n**, *som igjen multipliseres med 3, og dette resultatet adderes igjen med 19.*

Resultatet av et AR-direktiv skrives der direktivet står, på samme måte som resultatet av bruker-makro-kall.

Se ellers eksempel om FIG-CP på side 79 der AR-direktivet er brukt.

9.3 DIREKTIVET IF

IF-direktivet brukes for å teste logiske uttrykk. Direktivet består av fire deler: *Navn*, *Vilkår*, *THEN (Sann)*- og *ELSE (Usann)-stamme*, og det kalles på følgende måte:

^IF/Vilkår/THEN/ELSE;

der ELSE-stammen kan utelates.

Vilkår er et logisk uttrykk der følgende operatorer kan benyttes:

AND : Logisk relasjon OG.

OR : Logisk relasjon ELLER.

NOT : Negasjon.

MISD: Macro IS Defined (Makro er definert).

MIND: Macro Is Not Defined (Makro er ikke definert).

ODD : Test for oddetall.

EVEN: Test for partall.

' : Apostrof for strengsammenligning.

( ) : Parenteser for isolering av deler av logiske uttrykk.

= : Lik.

<> : Forskjellig fra.

>< : Forskjellig fra.

< : Mindre enn.

> : Større enn.

<= : Mindre enn eller lik.

=< : Mindre enn eller lik.

>= : Større enn eller lik.

=> : Større enn eller lik.

For *THEN* og *ELSE* er regelen den samme som for bruker-makro-stammer.

Vilkåret kan være enten *SANT* eller *USANT*. Dersom det er sant vil THEN-stammen bli ekspandert. Dersom det er usant, vil ELSE-stammen bli ekspandert.

Eks-29: *ˆIF/odd{ˆ$pn;}/odde nummer/par nummer;*

I dette kallet testes det hvorvidt den aktuelle siden har likt eller ulikt nummer. Hvis den har ulikt nummer, skrives teksten **odde nummer** *ut. I motsatt fall skrives teksten* **par nummer** *ut.*

På denne siden vil kallet gi som resultat: par nummer

I det neste eksemplet forutsetter vi at vi allerede har definert en makro med navn *DAG* som gir dagens navn som resultat.

Eks-30: *ˆIF/{ˆ$DD; = 20} AND*
{''ˆDAG;' <> 'Lørdag'} AND
{''ˆDAG;' <> 'Søndag'}/Lønningsdag!/Ikkeno' penger å få;

I dette kallet testes det på hvorvidt det er lønningsdag. Kriteriene for dette er at datoen skal være den 20., og dagen forskjellig fra lørdag og søndag. Da dette eksemplet ble laget ga det følgende resultat:

Ikkeno' penger å få

Det gjelder for *IF* som for alle andre direktiv/makro-kall, at hele kallet leses før det behandles. Det betyr i dette tilfellet at vilkåret ikke blir testet før hele kallet er innlest. Dette kan ha betydning ved sammenligning av strenger (tekstbiter).

Dersom vi f.eks. har en makro, *NAVN*, som inneholder teksten *Allison's*, og denne skal brukes i en strengsammenligning, må dette utføres på følgende måte:

Eks-31: *ˆIF/ˆ<'ˆNAVN;' = 'Lloyds'ˆ>/.....*

Etter at IF er lest inn, vil vilkåret se ut som følger:

'ˆNAVN;' = 'Lloyds'

Dette betyr at **ˆNAVN;** *blir ekspandert ved analysering av vilkåret. Dette fører igjen til at apostrofen som finnes inni makroen* **NAVN** *forekommer på et nivå høyere enn de apostrofene som omslutter* **NAVN**, *og vil derfor ikke forveksles med disse.*

Dersom uttrykket hadde vært skrevet uten sitattegn på følgende måte:

ˆIF/'ˆNAVN;' = 'Lloyds'/.....

ville vilkåret sett slik ut etter innlesing av IF-kallet:

'Allison's' = 'Lloyds'

Dette fører altså til at apostrofen som var inni **NAVN** *kommer på samme nivå som apostrofene som omslutter* **NAVN**, *og kan derfor ikke skilles fra disse. Dette vil derfor føre til mistolking av uttrykket, med en feilmelding som følge.*

Se ellers eksemplene om *FIG-CP* og *SIP* i kapittel 10 der IF-direktivet er brukt.

9.4 BRUK AV MAKRO-BIBLIOTEKER MED DIREKTIVET LIB-CONT

Ethvert program har en begrenset lagringskapasitet. For NOTIS-TF betyr dette at det er en grense for hvor mange makroer som kan defineres. Denne grensen er mer avhengig av summen av makroenes størrelse enn antall makroer. For å omgå dette problemet er det mulig å legge makro-definisjoner i biblioteksfiler. I denne forbindelse benyttes direktivet *LIB-CONT*.

LIB-CONT-direktivet kalles på følgende måte:

^LIB-CONT/Navn-1/Navn-2/Navn-3/..../Navn-n;

der hele kallet må skrives på en linje.

Når et makro-bibliotek inkluderes, leses det i første omgang fram til siste LIB-CONT-direktiv. LIB-CONT-direktivene trenger ikke nødvendigvis ligge først i biblioteksfilen, men alle LIB-CONT-direktivene må komme fortløpende.

En normal oppbygning av et makro-bibliotek vil derfor være:

Definering av små globale hjelpemakroer, samt integer-makroer, etc.

Deretter følger LIB-CONT for alle større makroer.

Tilslutt kommer definisjonen av de store makroene.

Når så et slikt bibliotek blir inkludert, leses alle definisjoner fram til første LIB-CONT. Deretter leses alle LIB-CONT-kallene, og lesingen stopper så der i første omgang.

Bruken av LIB-CONT fører til at makronavnene som angis som parametre til direktivet blir definert som brukermakroer, men selve stammen (som er den delen som er plasskrevende) forblir udefinert.

Når så en av makroene definert i LIB-CONT kalles, vil NOTIS-TF fortsette å lese biblioteksfilen inntil definisjonen av denne makroen finnes. Dette betyr altså at man kan ha tilgang til mange makroer, men det er kun de makroene som brukes som opptar plass i maskinens hukommelse.

Et makro-bibliotek inkluderes på vanlig måte ved hjelp av direktivet *^IN/Navn/L;*. L'en tilslutt her gir beskjed til NOTIS-TF om at filen skal behandles som et bibliotek (Library).

Dersom man ønsker å utvide standardbiblioteket som leveres med NOTIS-TF med flere makro-definisjoner, er det et godt råd å lage disse i et eget bibliotek, som så inkluderes i standardbiblioteket. Denne inkluderingen må gjøres før første LIB-CONT. Ved å gjøre det på denne måten kan man lett oppgradere nye versjoner av standardbiblioteket ved kun å inkludere sine egne makro-biblioteker.

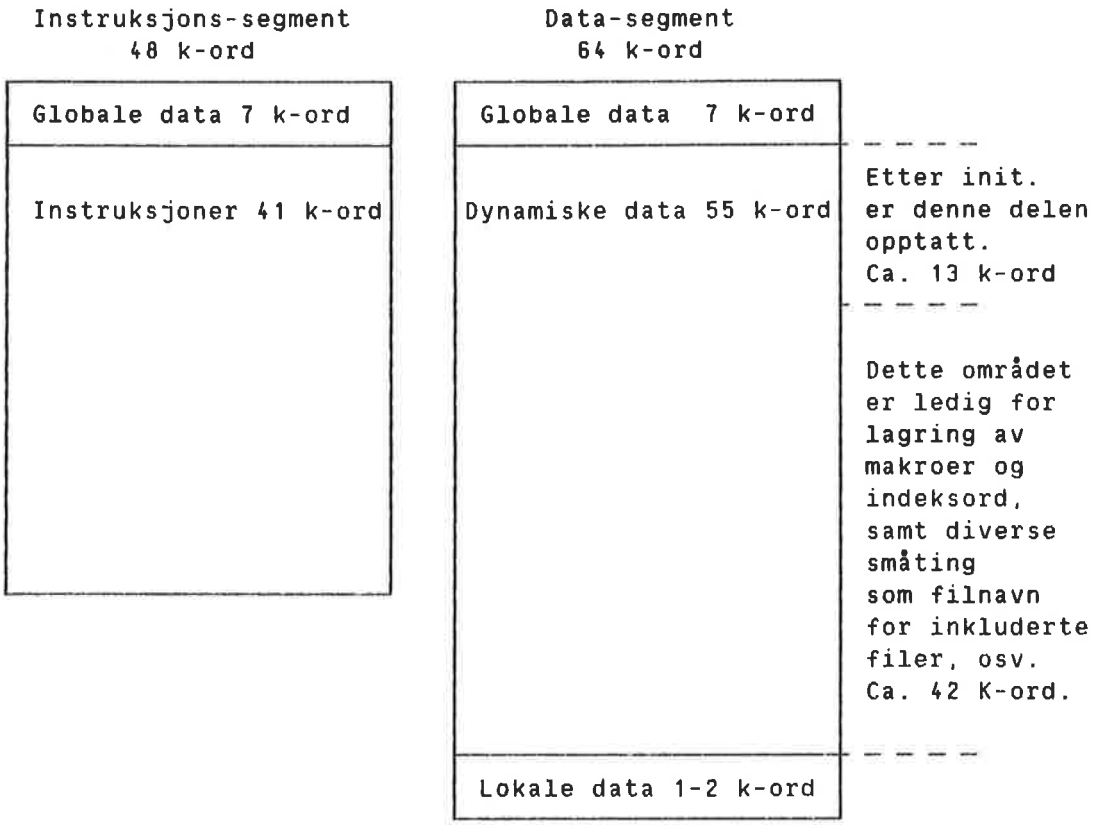
Den praktiske grensen for hvor mange biblioteker man kan inkludere vil ligge rundt 10, litt avhengig av hvor dype inkluderingsnivåer man ellers bruker i tekstdokumentene sine.

9.5 NOTIS-TF - RESSURS- OG PLASSFORBRUK I MASKINEN

Denne seksjonen gir en utdypning av hvordan NOTIS-TF fungererer internt, og man bør ha en del kjennskap til databehandling for å få utbytte av den. Det er ikke nødvendig å lese denne seksjonen for andre enn spesielt interesserte.

NOTIS-TF er en typisk "batch"-orientert applikasjon. Dette betyr at den utfører sin oppgave med liten eller ingen interaktiv påvirkning fra brukeren. Dette betyr videre at den ikke er avhengig av å vente på noen ting mens den arbeider, og det er derfor naturlig at den bruker alle maskinens ledige ressurser. Dette kan slå uheldig ut for interaktive brukere som konkurrerer med NOTIS-TF om maskinens ressurser.

NOTIS-TF eksekveres som et "to-bank" system i ND-10/ND-100 og ND-SATELLITE. Dette betyr at den bruker ett segment for instruksjoner og ett segment for data. Denne inndelingen kan illustreres på følgende måte (alle verdier er her tilnærmet):



Som det fremgår av illustrasjonen er det den gjenværende plassen av det dynamiske dataområdet som brukes til lagring av makroer og indeks-ord. Den opptatte delen av dette dynamiske dataområdet er benyttet til lagring av alle direktiver samt alle makro-hoder som er definert i biblioteket. Dette betyr at dersom man utvider biblioteket eller inkluderer flere biblioteker, vil denne delen øke.

For å gi deg et bildet av hvor stor plass de enkelte delene opptar, vil vi her beskrive de mest aktuelle dataelementene.

Det grunnleggende dataelementet kalles Basic-String-Unit (BSU). Dette er en enhet som benyttes for å lagre alle slags tegnstrenger, slik som makro-navn, makro-stamme, indeks-ord, filnavn, osv.

BSU 7 ord

1 BSU kan lagre opptil 8 tegn.

Som tidligere beskrevet består en makro av et hode, et navn og en stamme. Dette hodet er også et dataelement, Makro-Descriptor-Block (MDB).

MDB 30 ord

Et indeks-ord består av et hode pluss selve ordet, Index-Descriptor-Block (IDB).

IDB 10 ord

Enhver fil som håndteres av TF får tildelt et filhode, File-Descriptor-Block (FDB). Det vil ikke bli allokert flere FDB'er enn det er samtidig åpne filer. Dvs. at dersom det inkluderes flere filer på samme nivå, vil disse benytte samme FDB.

FDB 200 ord

Ut fra denne forklaringen vil vi nå vise noen eksempler på hvordan man kan beregne plassbehovet for forskjellige funksjoner.

Dersom man definerer en makro som følger:

```
^MD/TITTEL/Dette er tittelen;
```

vil denne oppta følgende plass:

```
1 MDB                          =  30 ord
Navn   =  6 tegn ): 1 BSU      =   7 ord
Stamme = 17 tegn ): 3 BSU      =  21 ord
-------------------------------------------
Totalt plassbehov              =  58 ord
```

==========

Dersom man definerer et indeksord som følger:

^XA/Langsgående skille;

vil dette oppta følgende plass:

```
1 IDB                          =  10 ord
Navn = 18 tegn ): 3 BSU        =  21 ord
-----------------------------------------
Totalt plassbehov              =  31 ord
=========================================
```

Som tidligere nevnt er alle disse tallene tilnærmede verdier, og må derfor brukes med en del slingringsmonn.

I denne sammenhengen vil vi også utdype virkningen av LIB-CONT-direktivet som brukes i makro-biblioteker. Det som skjer er at NOTIS-TF leser filen fram til og med siste LIB-CONT direktiv. Alle makroer som finnes i disse LIB-CONT vil da være definert, men de vil kun oppta 1 MDB pluss navnet i hukommelsen. Når de så brukes vil selve stammen bli lest inn fra filen. Dette gjør at man kan definere mange store makroer, men kun bruke noen få av dem samtidig.

Dersom man sletter en makro, vil hele den plassen den opptar bli frigjort slik at den kan benyttes igjen for neste makro som blir definert.

10 EKSEMPLER

I dette kapitlet vil vi ved hjelp av eksempler forsøke å illustrere hvordan makroer kan nyttes i praktisk bruk.

Kompleksiteten i makroene vil gradvis øke gjennom kapitlet.

De fleste eksemplene vil bygge på makroer som finnes i NOTIS-TFs makro-bibliotek.

10.1 MAKROEN "BOLD"

Denne bruker-makroens funksjon er å skrive en tekst i uthevet skrift.

Et kall til denne makroen kan f.eks. se ut som følger:

... noe som er ˆBOLD/viktig;.

Dette vil gi resultatet:

... noe som er ***viktig****.*

Definisjon: *ˆMD/BOLD/ˆ<ˆBT=+;ˆ1;ˆBT=-;ˆ>;*

Det første kallet som forekommer i stammen er *[ˆBT=+;]*. Dette slår på modus for uthevet skrift.

Deretter inkluderes en parameter. Det er teksten i denne parameteren som skal skrives med uthevet skrift.

Til slutt slås modus for uthevet skrift av igjen med kallet *[ˆBT=-;]*.

10.2 MAKROEN "UTHEV"

Engelsk navn: SHOW

Denne makroen har tildels samme funksjon som BOLD, men i tillegg til å skrive teksten i uthevet-skrift, vil den også bli skrevet sentrert på en egen linje.

Et kall til denne makroen kan f.eks. være:

... dette fører til at beløpet ˆUTHEV/kr. 10300; trekkes ....

Som gir resultatet:

... dette fører til at beløpet

kr. 10300

trekkes ....

Definisjon: *ˆMD/ˆ<ˆBL=1;ˆNF;ˆBT=+;ˆJM=C;ˆ1, ;ˆBL=1;ˆOF;ˆ>;*

Det første kallet, *[ˆBL=1;]*, vil gi en blank linje.

Kallet *[ˆNF;]* fører til at formatparametrene lagres.

Kallet *[ˆBT=+;]* slår på modus for uthevet tekst.

Kallet *[ˆJM=C;]* setter justerings-modus til sentrering. Dette fører til at etterfølgende tekst blir sentrert.

Deretter inkluderes en parameter. Denne parameterens standardverdi er satt til en blank. Dette er strengt tatt ikke nødvendig.

Etter parametren følger så kallet *[ˆBL=1;]*, som fører til at det blir skrevet en blank linje etter den sentrerte parameteren.

Til slutt kommer kallet *[ˆOF;]*. Dette fører til at sist lagrede formatparametre hentes fram igjen. I dette tilfellet betyr det at uthevet-skrift-modus og justerings-modus vil bli satt tilbake til de verdier de hadde før makroen ble ekspandert.

10.3 EKSEMPEL-MAKROEN I DENNE VEILEDNINGEN

I denne veiledningen har vi brukt en kombinasjon av integer-makroer og andre makrotyper for å få til automatisk nummerering av eksemplene.

For dette formålet har vi definert følgende makroer:

```
1) ^IM/EKS-NO/0;
2) ^MD/START-EKS/^<^EKS-NO=+1;^NF;^LM=+10;^FM=N;
   ^IP=Eks-^$EKS-NO;: ;^>;
3) ^MD/SLUTT-EKS/^<^OF;^>;
```

Først definerte vi en integer-makro med navn *EKS-NO* og verdi *0*.

Deretter definerte vi en brukermakro med navn *START-EKS*. Den virker på følgende måte: Først økes eksempelnummeret med 1: *[^EKS-NO = +1;]*. Deretter lagres alle formatparametre: *[^NF;]*. Så økes venstre marg med 10 posisjoner og fyllingsmodus settes til ikke-fyll: *[^LM=+10;^FM=N;]*. Deretter påbegynnes et innrykket avsnitt med teksten *Eks-n:*, der *n* er aktuelt eksempelnummer: *[^IP=Eks-^$EKS-NO;: ;]*.

Tilslutt definerte vi makroen *SLUTT-EKS*, som brukes for å avslutte et eksempel. Denne henter tilbake de lagrede formatparametrene: *[^OF;]*.

Et eksempel lages da på følgende måte:

```
Eks-32: ^START-EKS;Dette er et eksempel.......

        ...............
        ................

        ^SLUTT-EKS;
```

10.4 MAKROEN "BM"

Denne makroen brukes for å sette begge marger (LM og RM) samtidig.

Definisjon: *^MD/BM/^<^LM=^1,0;;^RM=1,0;;^>;*

Som du ser settes begge marger til samme verdi. Standardverdien er 0. Dvs. at dersom BM kalles uten parameter, vil begge marger få verdiene 0.

Det er viktig å merke seg at høyre-marg regnes fra høyre kant.

10.5 MAKROEN "CE"

Denne makroen brukes til å sentrere en tekst på en egen linje, evt. flere, avhengig av tekststørrelsen.

Et kall til denne makroen kan f.eks. være:

ˆCE/Denne teksten skal sentreres på en egen linje;

Som vil gi resultatet:

Denne teksten skal sentreres på en egen linje

Definisjon: *ˆMD/CE/ˆ<ˆNF;ˆBL;ˆJM=C;ˆ1;ˆBL;ˆOF;ˆ>;*

Det første kallet, *[ˆNF;]*, fører til at formatparametrene blir lagret unna, slik at de kan settes tilbake til opprinnelige verdier etter at makroen er ekspandert.

Det neste kallet, *[ˆBL;]*, fører til at en evt. ikke-avsluttet tekstlinje avsluttes. Dersom forrige tekstlinje er avsluttet, har dette kallet ingen virkning.

Deretter følger kallet *[ˆJM=C;]*, som setter justering-modus til sentrering. Dette fører til at etterfølgende tekst vil bli sentrert.

Så inkluderes en parameter. Dette skal da være den teksten som skal sentreres.

Den tekstlinjen som er lagd av den inkluderte parameter blir så avsluttet med kallet *[ˆBL;]*.

Og til slutt settes evt. modifiserte formatparametre tilbake til sine opprinnelige verdier ved hjelp av kallet *[ˆOF;]*. I dette tilfellet gjelder det justerings-modus.

10.6 MAKROEN "CP"

Denne makroen utfører et betinget sideskift. Dette betyr at dersom antall gjenværende linjer på siden er mindre enn et spesifisert antall, skal det foretas et sideskift. Hvis ikke skal aktuell tekstlinje avsluttes.

Definisjon: *^MD/CP/^<^IF/^AR=^$PL;-^$CLINE;-^$BB;+1; < ^1;*
/^<^PG;^>/^<^BL;^>;^>;

I første linje settes det opp et vilkår ved hjelp av et kall til IF-direktivet. Dette vilkåret består i å teste hvorvidt antall gjenværende linjer på siden er mindre enn det oppgitte antall.

Antall gjenværende linjer regnes ut, ved hjelp av AR direktivet (se side 54), på følgende måte:

```
  Sidelengde              ($PL    )
- Aktuellt linjenummer    ($CLINE)
- Nedre-kant              ($BB    )
+ 1
-------------------------------------
= Antall gjenværende linjer på siden
=====================================
```

Dette testes så mot det nummer som inkluderes som parameter 1.

Dersom vilkåret tilfredsstilles, utføres første-stamme til IF-direktivet, som i dette tilfellet er *[^PG;]*, hvilket gir sideskift.

Dersom vilkåret ikke tilfredsstilles utføres andre-stamme til IF-direktivet, som i dette tilfellet er *[^BL;]*, hvilket avslutter aktuell tekstlinje. Dersom aktuell tekstlinje er avsluttet, har det ingen virkning.

10.7 MAKROEN "SPRE"

Engelsk navn: SPREAD

Denne makroen skriver tre tekstbiter på en og samme linje, henholdsvis venstre-, senter- og høyre-justert, etterfulgt av et spesifisert antall blanke linjer.

Et kall til denne makroen kan f.eks. være:

ˆSPRE/HANSEN/JENSEN/OLSEN/2;

Som gir resultatet:

HANSEN *JENSEN* *OLSEN*

Definisjon:
```
ˆMD/SPRE/ˆ<ˆBL;ˆNF;ˆFM=F;ˆJM=L;ˆ1;ˆBL=-1;
ˆJM=C;ˆ2;ˆBL=-1;ˆJM=R;
ˆIF/''ˆ3,*-*;'<>'*-*'/ˆ3;/ˆ<ˆBL=1;ˆ>;
ˆBL=ˆ4,0;;ˆOF;ˆ>;
```

Kallet *[ˆBL;]* fører til at en evt. påbegynt tekstlinje avsluttes. Dersom ingen linje er påbegynt, har dette kallet ingen virkning.

Deretter lagres alle formatparameter verdier unna ved hjelp av kallet *[ˆNF;]*.

Fyllings-modus settes til full-fylling ved hjelp av kallet *[ˆFM=F;]*. Dette gjøres for at evt. linjeskift i parametre eller makro-stamme skal bli ignorert.

Justerings-modus settes til venstre-justering med kallet *[ˆJM=L;]*.

Deretter inkluderes parameter 1, og formateres venstre-justert i linjen.

Tekstlinjen avsluttes så med kallet *[ˆBL=-1;]*. Dette kallet fører til at det som befinner seg i aktuell tekstlinje skrives ut, og samme linje påbegynnes på nytt. Dvs. det blir ikke foretatt noe linjeskift.

Så settes justerings-modus til sentrering med kallet *[ˆJM=C;]*.

Og parameter 2 inkluderes. Teksten i denne parameteren vil bli sentrert i den samme linjen som parameter 1 ble venstre-justert.

Parameter 2 avsluttes på samme måte som parameter 1, med et *[ˆBL=-1;]* kall.

Justerings-modus settes så til høyre-justering med kallet *[^JM=R;]*. Slik blir parameter 3 høyre-justert.

Det skal ikke være nødvendig å ha med noen parameter 3 til dette kallet. Derfor blir det testet på hvorvidt denne er spesifisert i kallet. Denne måten å teste på finner du igjen i flere makroer i makro-biblioteket. Testen foretas med IF-direktivet og utføres på følgende måte:

> Inkluderingen av parameter 3 skrives mellom enkle apostrofer. Dette gjøres for å angi at det er en streng-sammenlikning.
>
> Parameteren inkluderes med standardverdi "*-*". Dette betyr at dersom parameteren ikke er spesifisert i kallet, vil den første strengen i sammenlikningen være "*-*". Legg også merke til at inkluderingen av parameter 3 IKKE er skrevet i sitattegn. Hvis så hadde blitt gjort, ville det betydd at parameter 3 til IF-kallet skulle inkluderes.
>
> Sammenlikningen skal foretas med operatoren "<>" (forskjellig fra).
>
> Den andre strengen i sammenlikningen er "*-*".
>
> Dette resulterer i at dersom parameter 3 er spesifisert, og har verdi forskjellig fra "*-*", vil resultatet av sammenlikningen bli SANN. I motsatt tilfelle blir sammenlikningen mellom to like strenger, med resultatet USANN.

Dersom resultatet av sammenlikningen blir SANN, vil første-stamme til IF-direktivet bli utført. Dvs. at parameter 3 vil bli inkludert. Legg her merke til at inkluderingen av parameter 3 IKKE står i sitattegn. Hvis så hadde vært tilfelle, ville det betydd at parameter 3 til IF-kallet skulle inkluderes, noe som jo ikke er ønskelig.

Dersom resultatet av sammenlikningen er USANN, vil andre-stamme til IF direktivet bli utført. Dvs. at det vil bli skrevet en blank linje.

Deretter skrives det antall blanke linjer som er spesifisert i parameter 4, standard er 0.

Til slutt settes formatparametrene tilbake til sine opprinnelige verdier ved hjelp av ^OF-kallet.

10.8 MAKROEN "FIG-CP"

Denne makroen setter av plass til en figur, og skriver en figur-tekst med løpende figurnummer under figuren. Dersom det ikke er plass til figuren i den gjenværende plassen på siden, flyttes figuren til neste side, mens teksten fortsetter på den siden man er. Figuren blir altså flyttet i forhold til teksten, men evt. referanser til figurnummeret opprettholdes. Dersom figuren er større enn den maksimale side-størrelsen gis det en feilmelding på skjermen under formateringen, og det skrives en feilmelding i teksten der hvor man forsøkte å plassere figuren.

Et kall til denne makroen kan f.eks være:

^FIG-CP/7/Her kunne det ha stått en figur;

som gir resultatet:

Fig. 1. Her kunne det ha stått en figur

Definisjon:

```
^MD`FIG-CP`
^<^IM/FIG-ROM/^1,20;;^MD`FIG-TEKST`^PM,2;;^FN=+1;^BL;
  ^IF|^AR=^$FIG-ROM;+4;>^AR=^$PL;-^$TB;-^$BB;;|
  ^<^MS`** Feil i bruk av makro FIG-CP, fig. ^$FN;, -for stor figur **;
    ^BL=2;^NF;^BM=0;
    ^JM=c;_Feil i bruk av makro FIG-CP, fig. ^$FN;, -for stor figur_
    ^BL=2;^OF;
  ^>|
  ^<^IF|(^AR=^$FIG-ROM;+4;>^AR=^$PL;-^$CLINE;+1-^$BB;;)OR(^$FIG-PN;=^$PN;)|
    ^<^PH-STR=^<^FIG-CP!^>^$FIG-ROM;!^FIG-TEKST;^<;^>;
      ^FIG-VENT=+1;^FIG-PN=^$PN;;^RYDDOPP;^>|
      ^<^IF|^$FIG-VENT;>0|^<^FN=^$FN;-^$FIG-VENT;;^FIG-VENT=0;^>;
        ^BL=^$FIG-ROM;+1;
        ^NF;^BM=0; ^JM=C;_Fig. ^$FN;. ^FIG-TEKST;_^BL=2;^OF;
      ^>;
    ^>;
  ^MK/FIG-TEKST;^MK/FIG-ROM;
^>;
```

Makroen kan se veldig komplisert og skremmende ut ved første øyekast, men ved å dele den opp i seksjoner er det lettere å få oversikt. Vi kan derfor dele den opp i 5 logiske seksjoner:

1

```
^MD`FIG-CP`
^<^IM/FIG-ROM/^1,20;;^MD`FIG-TEKST`^PM,2;;^FN=+1;^BL;
```

Denne seksjonen foretar de nødvendige initieringer, som å definere hjelpemakroer, osv.

2

```
  ^IF|^AR=^$FIG-ROM;+4;>^AR=^$PL;-^$TB;-^$BB;;|
  ^<^MS`** Feil i bruk av makro FIG-CP, fig. ^$FN;, -for stor figur **;
    ^BL=2;^NF;^BM=0;
    ^JM=c;_Feil i bruk av makro FIG-CP, fig. ^$FN;, -for stor figur_
    ^BL=2;^OF;
  ^>|
```

Denne seksjonen utfører testing, og feilhåndtering dersom figuren er større en det maksimalt mulige.

3

```
  ^<^IF|(^AR=^$FIG-ROM;+4;>^AR=^$PL;-^$CLINE;+1-^$BB;;)OR(^$FIG-PN;=^$PN;)|
    ^<^PH-STR=^<^FIG-CP!^>^$FIG-ROM;!^FIG-TEKST;^<;^>;
      ^FIG-VENT=+1;^FIG-PN=^$PN;;^RYDDOPP;
    ^>|
```

Denne seksjonen utfører testing på hvorvidt det er plass til figuren i aktuell side, hvis ikke legger den figuren i kø for neste side.

4

```
    ^<^IF|^$FIG-VENT;>0|^<^FN=^$FN;-^$FIG-VENT;;^FIG-VENT=0;^>;
      ^BL=^$FIG-ROM;+1;
      ^NF;^BM=0; ^JM=C;_Fig. ^$FN;. ^FIG-TEKST;_^BL=2;^OF;
    ^>;
```

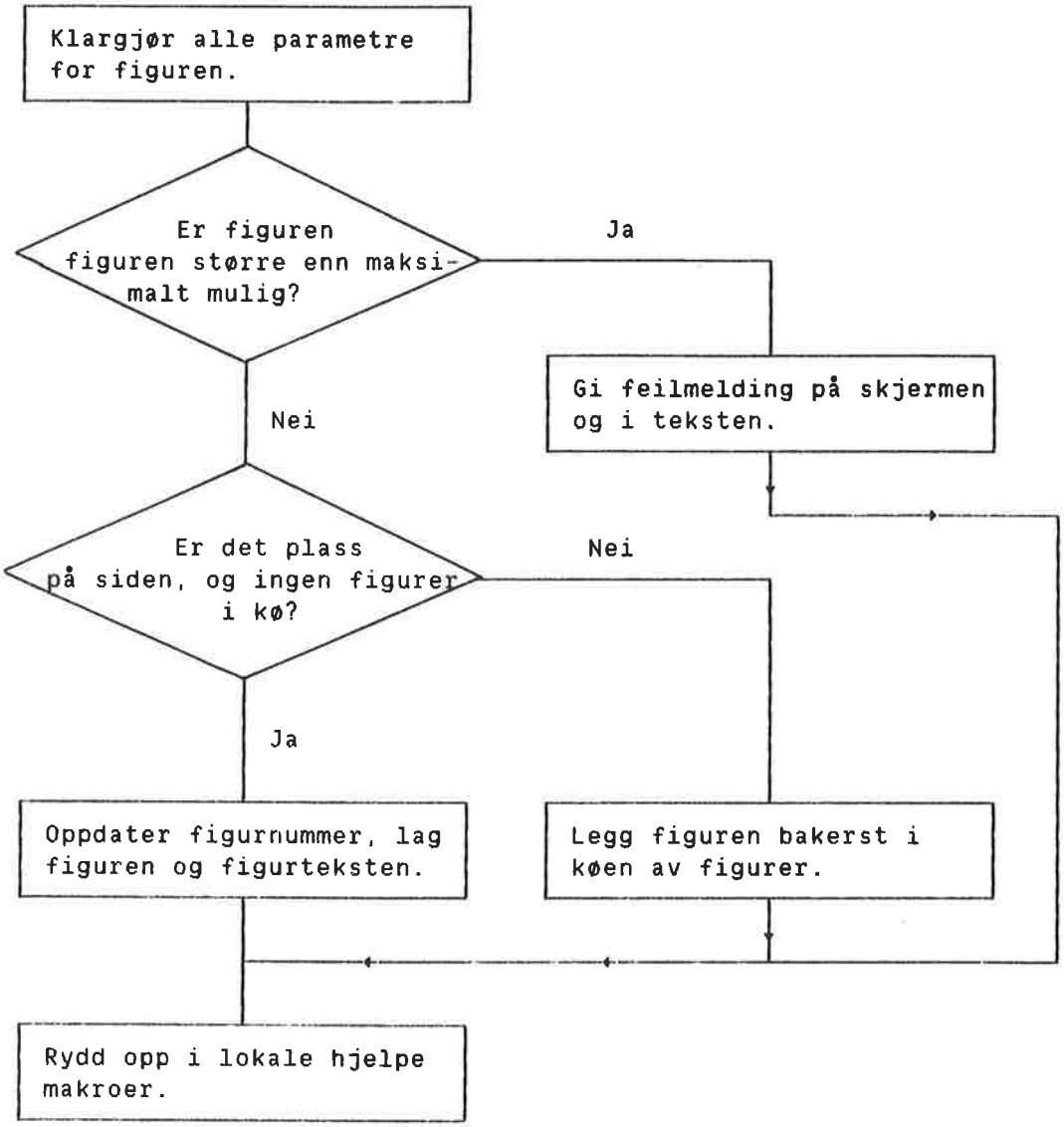
Denne seksjonen utfører selve figur-håndteringen, dvs. setter av spesifisert plass.

5

```
  ^>;
  ^MK/FIG-TEKST;^MK/FIG-ROM;
^>;
```

Denne seksjonen utfører nødvendig opprydding, slik som å fjerne lokale hjelpemakroer.

Et flytskjema over makroen kan vel også være med på å øke forståelsen:



Nå som hver seksjon har fått beskrevet sin logiske funksjon skal vi detaljbeskrive hver enkelt seksjon noenlunde separat. Men før vi gjør dette må de impliserte globale hjelpemakroer beskrives:

FN : Dette er en integer-makro som inneholder løpende figur-nummer.

FIG-VENT: Dette er en integer-makro som inneholder antall figurer som ligger i kø for neste side.

FIG-PN : Dette er en integer-makro som inneholder sidetallet for siden hvor den figuren som ligger sist i køen var forsøkt laget.

RYDDOPP : Dette er en makro som definerer operasjoner som skal utføres for å tømme en evt. figurkø etter siste side.

DETALJBESKRIVELSE AV SEKSJON 1 I FIG-CP MAKROEN

Først defineres integer-makroen FIG-ROM til å inneholde verdien av parameter-1 til FIG-CP-kallet, antall linjer som skal avsettes til figuren, der standardverdien er 20: *[^IM=FIG-ROM=^1,20;;]*.

Deretter defineres makroen FIG-TEKST til å inneholde parameter-2 til FIG-CP-kallet, figurteksten *[^MD`FIG-TEKST`^PM,2;;]*. Her har vi brukt PM-direktivet fordi vi gjerne vil ha muligheten til å inkludere et direktiv i parameter 2, f.eks. et direktiv for venstrejustering av teksten.

Så telles det fortløpende figurnummeret opp med en *[^FN=+1;]* slik at den nye figuren skal få ett nummer høyere enn forrige figur.

Til slutt avbrytes aktuell tekstlinje *[^BL;]*. Dersom aktuell tekstlinje er tom, har dette direktivet ingen virkning.

DETALJBESKRIVELSE AV SEKSJON 2 I FIG-CP MAKROEN

Her testes det *[^IF]* hvorvidt figuren pluss 4 linjer *[^$FIG-ROM;+4]* er større en det antall linjer som er tilgjengelig på en tekstside *[^AR.^$PL;-^$TB;-^$BB;;]*. Tallet for antall linjer som er tilgjengelige regnes altså ut som sidelengde minus øvre- og nedre kant. Som nevnt testes dette tallet mot figurstørrelsen pluss 4 linjer. Disse 4 linjene er nødvendige for å få plass til figurteksten, samt blank linje før og etter denne.

Dette IF-direktivet har en THEN- og en ELSE-del. THEN-delen, som utføres dersom testen gir verdien SANN, inneholder resten av seksjon-2. ELSE-delen, som utføres dersom testen gir verdien USANN, inneholder seksjon 3 og 4. Dette betyr altså at dersom figuren er for stor for en side, SANN verdi, utføres resten av seksjon 2 og seksjon 5. I motsatt tilfelle utføres seksjon 3, evt. 4, og seksjon 5.

Utførelse av resten av seksjon 2 fører til at det blir skrevet en melding på skjermen *[^MS]*, inneholdende figurnummeret *[^$FN;]*. Den samme meldingen blir også skrevet i teksten på det punkt hvor man forsøkte å avsette plass til figuren. Meldingen i teksten blir innledet med to blanke linjer *[^BL=2;]*. Formatparametrene blir lagret *[^NF;]*. Margene blir satt til null *[^BM=0;]*. Justeringsmodus blir satt til sentrering *[^JM=C;]*. Meldingen, inneholdende figurnummeret *[^$FN;]*, blir så skrevet understreket og sentrert på en linje. Meldingen blir avsluttet med to blanke linjer *[^BL=2;]*. Og til slutt blir formatparametrene satt tilbake til sine opprinnelige verdier *[^OF;]*,i dette tilfellet margene.

DETALJBESKRIVELSE AV SEKSJON 3 I FIG-CP MAKROEN

Denne seksjonen innledes også med en test *[^IF]*. Her testes det hvorvidt figuren pluss 4 linjer *[^$FIG-ROM;+4]* er større enn det antall linjer som er igjen på aktuell tekstside *[^AR.^$PL;-^$CLINE;+1-^$BB;;]*. Tallet for antall gjenværende linjer utregnes altså som sidelengde minus aktuell linje pluss 1 minus nedre kant. Som nevnt testes dette tallet mot figurstørrelsen pluss 4 linjer. Disse 4 linjene er nødvendige for å få plass til figurteksten, samt blank linje før og etter denne. Dette IF-direktivet inneholder faktisk to tester med logisk relasjon ELLER. Dette betyr at dersom en av testene gir verdien SANN, blir det totale resultatet SANT. Den andre testen består i å sjekke om det er forsøkt å avsette plass til figurer før på aktuell side som har blitt lagt i kø for neste side *[^$FIG-PN;=^$PN;]*. Hvis så er tilfelle må også denne figuren legges i kø for neste side, uansett om det er plass til den eller ikke på aktuell side, slik at rekkefølgen av figurene ikke forandres.

Dette IF-direktivet har en THEN- og en ELSE-del. THEN-delen, som utføres dersom testen gir verdien SANN, inneholder resten av seksjon 3. ELSE-delen, som utføres dersom testen gir verdien USANN, inneholder seksjon 4. Dette betyr altså at dersom det ikke er plass til figuren på aktuell tekstside, SANN verdi, utføres resten av seksjon 3 og seksjon 5. I motsatt tilfelle utføres seksjon 4 og seksjon 5.

Utførelse av resten av seksjon 3 fører til at dette FIG-CP-kallet legges inn i pageheader-string *[^PH-STR]*, slik at det blir utført på nytt på toppen av neste side. Dersom det ligger FIG-CP-kall i pageheader-string allerede, blir dette kallet lagt i kø etter de andre. Videre økes tallet for antall figurer i kø med én *[^FIG-VENT=+1;]*. Og sidenummeret for siste figur i køen for neste side settes lik aktuell side *[^FIG-PN=^$PN;;]*. Til slutt i denne seksjonen kalles makroen RYDDOPP for å aktivisere en evt. opprydding av figur-køen etter siste side i dokumentet. Et kall til denne makroen har kun virkning første gangen, så gjentatte kall betyr ingenting.

DETALJBESKRIVELSE AV SEKSJON 4 I FIG-CP MAKROEN

Det er denne seksjon som setter av plass til figuren og skriver figurteksten.

Det innledes med en test *[^IF]* på hvorvidt det ligger figurer i kø *[^$FIG-VENT; > 0]*. Hvis så er tilfelle, så ligger også denne figuren i køen. Figurnummeret må derfor justeres med det antall figurer som venter i køen *[^FN=^$FN;-^$FIG-VENT;;]*. I tillegg må tallet for antall figurer i kø nullstilles *[^FIG-VENT=0;]*.

Det settes så av plass til figuren, pluss en blank linje før figurteksten *[^BL=^$FIG-ROM;+1;]*. Formatparametrene lagres *[^NF;]*. Margene settes til 0 *[^BM=0;]* slik at figurteksten skal bli midtstilt i linjen. Justeringsmodus settes til sentrering *[^JM=C;]*. Figurteksten inneholdende figurnummeret skrives ut understreket, etterfulgt av to blanke linjer *[^BL=2;]*. Og til slutt blir formatparametrene satt tilbake til sine opprinnelige verdier *[^OF;]*,i dette tilfellet margene.

DETALJBESKRIVELSE AV SEKSJON 5 I FIG-CP MAKROEN

Denne seksjonen utfører nødvendig opprydding. Dette består i å slette makroene FIG-TEKST *[^MK/FIG-TEKST;]* og FIG-ROM *[^MK/FIG-ROM;]*.

10.9 MAKROEN "SIP"

Denne makroen brukes til å innlede en sekvens av innrykkede avsnitt. Selve avsnittene innledes med makroen NP, som defineres inne i SIP. Sekvensen av innrykkede avsnitt skal avsluttes med EIP, som vil bli omtalt til slutt i denne seksjonen.

Makroen SIP kan kalles med opptil 5 parametre, som har følgende betydning:

1) Denne parameteren beskriver hvordan avsnitt laget med NP-makroen skal merkes. Alternativene er:

   N: Fortløpende nummerering med numerisk verdi.

   R: Fortløpende nummerering med romertall, store bokstaver.

   r: Fortløpende nummerering med romertall, små bokstaver.

   A: Fortløpende nummerering med store bokstaver.

   a: Fortløpende nummerering med små bokstaver.

   Dersom denne parameteren utelates, vil avsnitt laget med NP-makroen merkes med en strek "-" i margen.

2) Denne parameteren angir størrelsen på venstre marg for avsnittene. Dersom denne parameteren utelates, velges verdien 9.

3) Denne parameteren angir antall blanke linjer mellom hvert avsnitt. Dersom denne parameteren utelates, velges standardverdien for avsnittsmellomrom ($PS).

4) Denne parameteren angir hvordan tekst/merking i margen skal justeres. Alternativene her er de samme som for direktivet JM. Dersom denne parameteren utelates velges høyre-justering.

5) Denne parameteren angir hvordan teksten i avsnittene skal justeres. Alternativene her er de samme som for direktivet JM. Dersom denne parameteren utelates, brukes den verdien som gjelder til enhver tid.

Følgende eksempel er ment å illustrere noen av mulighetene ved bruk av SIP-, NP- og EIP-makroene.

Følgende sekvens av makroer og tekst:

```
^SIP/N/3;
^NP;Dette er første avsnitt nummerert med tall.
^NP;Dette er andre  avsnitt nummerert med tall.
^SIP/A/3;
^NP;Dette er første avsnitt nummerert med store bokstaver.
^SIP/r/5;
^NP;Dette er første avsnitt nummerert med romertall i små bokstaver.
^NP;Dette er andre  avsnitt nummerert med romertall i små bokstaver.
^EIP;
^NP;Dette er andre  avsnitt nummerert med store bokstaver.
^EIP;
^NP;Dette er tredje avsnitt nummerert med tall.
^EIP;
```

gir dette resultatet:

1) Dette er første avsnitt nummerert med tall.

2) Dette er andre avsnitt nummerert med tall.

A) Dette er første avsnitt nummerert med store bokstaver.

i) Dette er første avsnitt nummerert med romertall i små bokstaver.

ii) Dette er andre avsnitt nummerert med romertall i små bokstaver.

B) Dette er andre avsnitt nummerert med store bokstaver.

3) Dette er tredje avsnitt nummerert med tall.

Definisjon:

```
^MD/SIP/
^<^NF;^FM=C;
  ^IM=IPNO=1;          ^MD=HOW=^1,*-*;;  ^IM=MARG=^2,9;;    ^LM=+^$MARG;;
  ^IM=IPS=^3,^$PS;;;  ^MD=IJM=^4,R;;    ^MD=JUST=^5,*-*;;  ^BL;
  ^MD/IP/
  ^<^BL=^$IPS;;^CM/MD-SAVP;^MA/MD-SAVP/^PM=1;;
    ^IF|^$LM;<2|
    ^<^MS/** Feil i bruk av makro IP,^MD-SAVP;, LM må være >= 2. **;^>|
    ^<^CP=^$PF;;^NF;^JM=^IJM;;^IM-SAVP=^$LM;;
      ^IF|''^IJM;'='R'|
      ^<^LM=0;^RM=^$TW;-^$IM-SAVP;;^>|
      ^<^IF|''^IJM;'='L'|
        ^<^LM=-^$MARG;;^RM=0;^>|
        ^<^IF|('^IJM;'='C')OR('^IJM;'='S')|
          ^<^LM=-^$MARG;;^RM=^$TW;-^$IM-SAVP;;^>;
        ^>;
      ^>;
      ^MD-SAVP; ^IF|(^$RM;=0)AND(^$CPOS;>^$IM-SAVP;)|^<^BL;^>|^<^BL=-1;^>;
      ^OF;
    ^>;
  ^>;
  ^IF|''^HOW;'='*-*'|
  ^<^MD/NP/^<^IP=- ;^>;^>|
  ^<^MD/NP/^<^IP=^$IPNO,^HOW;;) ;^IPNO=+1;^>;^>;
  ^IF|''^JUST;'<>'*-*'|^<^JM=^JUST;;^>;
^>;
```

Denne makroen kan også deles opp i logiske seksjoner, for på den måten å lette oversikten. Vi har delt makroen opp i 4 hoved-seksjoner, hvor seksjon 2 i seg selv deles opp i 5 under-seksjoner.

1

```
^MD/SIP/
^<^NF;^FM=C;
  ^IM=IPNO=1;          ^MD=HOW=^1,*-*;;  ^IM=MARG=^2,9;;    ^LM=+^$MARG;;
  ^IM=IPS=^3,^$PS;;;  ^MD=IJM=^4,R;;    ^MD=JUST=^5,*-*;;  ^BL;
```

Denne seksjonen foretar de nødvendige initieringer, som å definere hjelpemakroer, osv.

2.1

```
  ^MD/IP/
  ^<^BL=^$IPS;;^CM/MD-SAVP;^MA/MD-SAVP/^PM=1;;
```

Hele seksjon 2 består i å definere makroen IP, som er basis for innrykkede avsnitt. Seksjon 2.1 foretar de nødvendige initieringer, som å ta vare på parametre, osv.

2.2

```
    ^IF|^$LM;<2|
    ^<^MS/** Feil i bruk av makro IP,^MD-SAVP;, LM må være >= 2. **;^>|
```

Denne seksjonen tester om hvorvidt venstre marg er stor nok for et innrykket avsnitt, og gir evt. feilmelding.

2.3

```
    ^<^CP=^$PF;;^NF;^JM=^IJM;;^IM-SAVP=^$LM;;
```

Denne seksjonen innleder selve avsnittet.

2.4

```
      ^IF|'^IJM;'='R'|
      ^<^LM=0;^RM=^$TW;-^$IM-SAVP;;^>|
      ^<^IF|'^IJM;'='L'|
        ^<^LM=-^$MARG;;^RM=0;^>|
        ^<^IF|('^IJM;'='C')OR('^IJM;'='S')|
          ^<^LM=-^$MARG;;^RM=^$TW;-^$IM-SAVP;;^>;
        ^>;
      ^>;
```

Denne seksjonen foretar testinger og justering av marger for avsnittet.

2.5

```
      ^MD-SAVP; ^IF|(^$RM;=0)AND(^$CPOS;>^$IM-SAVP;)|^<^BL;^>|^<^BL=-1;^>;
      ^OF;
    ^>;
  ^>;
```

Denne seksjonen lager det innrykkede avsnittet. Og avslutter IP-makroen.

3

```
  ^IF|''^HOW;'='*-*'|
  ^<^MD/NP/^<^IP=- ;^>;^>|
  ^<^MD/NP/^<^IP=^$IPNO,^HOW;;) ;^IPNO=+1;^>;^>;
```

Denne seksjonen definerer makroen NP, ut fra parametre spesifisert til SIP-kallet. NP-makroen er et alternativ til IP-makroen, ved at den gir en standard tekst i margen.

4

```
  ^IF|''^JUST;'<>'*-*'|^<^JM=^JUST;;^>;
^>;
```

Denne seksjonen tester og setter justeringsmodus for selve tekst-delen i de innrykkede avsnittene som skal følge. Samt at den avslutter SIP-makroen.

Før vi detaljbeskriver hver enkelt seksjon separat, vil vi beskrive de impliserte globale hjelpemakroer:

IM-SAVP : Dette er en integer-makro som brukes i IP-makroen for å ta vare på venstre marg for avsnittet, slik at denne kan brukes i beregninger og tester.

MD-SAVP : Dette er også en makro som brukes i IP-makroen for å ta vare på den tekst-parameteren som skal skrives i venstre marg.

DETALJBESKRIVELSE AV SEKSJON 1 I SIP MAKROEN

Denne seksjonen innledes med at alle formatparametre lagres *[^NF;]*. Deretter settes fyllingsmodus til betinget-fylling *[^FM=C;]*.

Det defineres så en integer-makro *[^IM=IPNO=1;]* med verdi 1. Denne makroen brukes for fortløpende nummerering av de innrykkede avsnittene ved bruk av NP-makroen.

Det defineres så en makro til å ta vare på parameter 1 til SIP-kallet *[^MD=HOW=^1,*-*;;]*.

Videre defineres det en integer-makro til å ta vare på parameter 2 til SIP-kallet, med standard-verdi 9 *[^IM=MARG=^2,9;;]*.

Venstre-marg økes med den verdi som ble lagret i MARG *[^LM=+^$MARG;;]*

Det defineres så en integer-makro til å ta vare på parameter 3 til SIP-kallet, hvor standard verdi settes til standard avsnitts-mellomrom *[^IM=IPS=^3,^$PS;;;]*.

Det defineres så en makro til å ta vare på parameter 4 til SIP-kallet, med standard-verdi *R [^MD=IJM=^4,R;;]*.

Det defineres så en makro til å ta vare på parameter 5 til SIP-kallet, med standard tom-parameter-verdi "*-*" som standard *[^MD=JUST=^5,*-*;;]*.

Til slutt avsluttes aktuell tekstlinje i dokumentet *[^BL;]*. Dersom det ikke finnes noen aktuell tekstlinje, har dette direktivet ingen virkning.

DETALJBESKRIVELSE AV SEKSJON 2.1 I SIP MAKROEN

Som tidligere nevnt består hele seksjon 2 i å definere IP-makroen. Denne seksjonen definerer innledningen i IP-makroen.

Først lages avsnittsmellomrom *[^BL=^$IPS;;]*

Deretter blankes den makroen som skal lagre marg-teksten for avsnittet *[^CM/MD-SAVP;]*, og parameter-1 til IP-kallet lagres i denne makroen *[^MA/MD-SAVP/^PM=1;;]*.

DETALJBESKRIVELSE AV SEKSJON 2.2 I SIP MAKROEN

Denne seksjonen består i å teste hvorvidt venstre-marg er stor nok til å lage et innrykket avsnitt. Minimumsverdien er her satt til 2 *[^IF|^$LM;<2|]*. Denne testen har en THEN- og en ELSE-del. Dersom testen gir resultatet SANN, dvs. at margen er for liten, vil resten av seksjon 2.2 bli utført. I motsatt tilfelle vil seksjonene 2.3, 2.4 og 2.5 bli utført.

Resten av seksjon 2.2 består i å skrive en feilmelding på skjermen *[^MS]*. Denne meldingen inneholder navnet til IP-makroen sammen med den parameteren den ble kalt med.

DETALJBESKRIVELSE AV SEKSJON 2.3 I SIP MAKROEN

Denne seksjonen starter med et betinget sideskift *[^CP=^$PF;;]*. Dette betyr at dersom det er færre linjer enn avsnitts-slutt ($PF) igjen på siden, vil det bli utført et sideskift.

Deretter lagres alle formatparametre *[^NF;]*.

Justeringsmodus settes til den verdien som er definert for justering av marg-teksten *[^JM=^IJM;;]*.

Og størrelsen på venstre-marg lagres i integer-makroen IM-SAVP *[^IM-SAVP=^$LM;;]*.

DETALJBESKRIVELSE AV SEKSJON 2.4 I SIP MAKROEN

Denne seksjonen innledes med å teste hvorvidt justerings-modus for marg-teksten er høyre (R) *[ˆIF|''ˆIJM;'='R'|]*. Dersom dette er tilfelle, utføres *kun* neste linje i denne seksjonen. I motsatt tilfelle utføres *ikke* neste linje, men linjene deretter.

I det tilfelle hvor marg-tekst-justeringen er høyre, blir venstre-marg satt til 0 *[ˆLM=0;]* og høyre-marg blir satt til den posisjonen venstre-marg hadde *[ˆRM=ˆ$TW;-ˆ$IM-SAVP;;]*. Dette betyr at tekstlinjen i sin helhet begrenses til å være den delen som er avsatt til marg-tekst.

Dersom marg-tekst-justeringen ikke er høyre, testes det hvorvidt den er venstre (L) *[ˆIF|''ˆIJM;='L'|]*. Dersom dette er tilfelle utføres *kun* neste linje i denne seksjonen. I motsatt tilfelle utføres *ikke* neste linje, men linjene deretter.

I det tilfelle hvor marg-tekst-justeringen er venstre, blir venstre-marg redusert med den størrelsen som er definert for marg-teksten *[ˆLM=-ˆ$MARG;;]* og høyre marg blir satt til 0 *[ˆRM=0;]*. Dette betyr at marg-teksten skal justeres mellom venstre-marg og høyre-kant av arket, dvs. linjenlengden til avsnitts-teksten pluss den delen som er avsatt til marg-tekst.

Dersom marg-tekst-justeringen heller ikke er venstre, testes det om den er sentrering (C) eller strekking (S) *[ˆIF|[''ˆIJM;'='C']OR[''ˆIJM;'='S']|]*. Dersom dette er tilfelle utføres resten av denne seksjonen. I motsatt tilfelle avsluttes denne seksjonen.

I det tilfellé hvor marg-tekst-justeringen er sentrering eller strekking, blir venstre-marg redusert med den størrelsen som er definert for marg-teksten *[ˆLM=-ˆ$MARG;;]* og høyre marg blir satt til den posisjonen venstre-marg hadde *[ˆRM=ˆ$TW;-ˆ$IM-SAVP;;]*. Dette betyr at tekstlinjen i sin helhet begrenses til å være den delen som er avsatt til marg-tekst.

DETALJBESKRIVELSE AV SEKSJON 2.5 I SIP MAKROEN

I seksjon 2.3 og 2.4 er de forskjellige formatparametre for justering av marg-teksten definert, denne seksjonen innledes derfor med å justere marg-teksten *[^MD-SAVP;]*.

Når dette er gjort testes det hvorvidt margteksten strekker seg over venstre marg for avsnitts-teksten. Dette kan bare skje dersom høyre-marg er 0, denne betingelsen er derfor tatt med i testen
[^IF|{^$RM;=0}AND{^$CPOS;>^$IM-SAVP;}].

Hvis testen gir verdien SANN, dvs. at marg-teksten strekker seg over venstre-marg for avsnittet, blir linjen avbrutt slik at avsnitts-teksten begynner i en ny linje *[^BL;]*.

I motsatt tilfelle avbrytes linjen slik at avsnitts-teksten kan begynne i samme linje *[^BL=-1;]*, men da selvfølgelig lenger til høyre.

Seksjonen, og hele IP-makroen, avsluttes så med at de format-parametrene som ble lagret i begynnelsen av seksjon 2.3 hentes fram igjen.

DETALJBESKRIVELSE AV SEKSJON 3 I SIP MAKROEN

Denne seksjonen innledes med at det testes hvorvidt det er spesifisert hvordan avsnitt laget ved hjelp av NP-makroen skal merkes *[ˆIF|''ˆHOW;'='*-*'|]*. Dersom denne testen gir verdien SANN, dvs. at merke-metode ikke er spesifisert, vil NP-makroen bli definert til å kalle IP-makroen med teksten '- ' *[ˆMD/NP/ˆ<ˆIP=- ;ˆ>;]*.

I motsatt tilfelle vil NP-makroen bli definert til å kalle IP-makroen med avsnitts-nummer ($IPNO) som parameter, skrevet i spesifisert format, etterfulgt av ') ' *[ˆMD/NP/ˆ<ˆIP=ˆ$IPNO,ˆHOW;;] ;]*, og NP-makroen vil øke avsnitts-nummeret med én *[ˆIPNO=+1;]*.

DETALJBESKRIVELSE AV SEKSJON 4 I SIP MAKROEN

Denne seksjonen tester hvorvidt det er spesifisert justeringsmodus for avsnitts-teksten *[ˆIF|''ˆJUST;'<>'*-*'|]*. Hvis så er tilfelle vil justeringsmodus bli satt til denne spesifiserte verdien *[ˆJM=ˆJUST;;]*.

En sekvens av innrykkede avsnitt innledet med SIP-makroen skal alltid avsluttes med EIP-makroen.

Definisjon:

^MD/EIP/^<^MK=IPNO;^MK=NP;^MK=HOW;^MK=JUST;^MK=IJM;
^LM=-^$MARG;;^BL=^$IPS;;^MK=MARG;^MK=IPS;^MK=IP;^OF;^>;

Ekspansjonen av denne makroen fører til at aktuell versjon av følgende hjelpemakroer definert i *SIP* slettes: *IPNO*, *NP*, *HOW*, *JUST*, *IJM*, *MARG*, *IPS* og *IP*. Margen settes tilbake til foregående verdi, det blir lagd et avsnittsmellomrom, og lagrede formatparametre blir hentet fram igjen.

Dette betyr at ved nestede SIP-nivåer slettes alle makro-definisjoner på aktuelt SIP-nivå, slik at definisjonene for foregående SIP-nivå blir gjeldende.

10.10 MAKROEN "HODE"

Denne makroen er ment å brukes til å lage en enkel utgave av firmaets logo for bruk i brev, notater, o.l. Makroen gjør bruk av en del globale makroer som er definert i makro-biblioteket, men som må modifiseres av hver enkelt brukerorganisasjon:

FIRMA	:	Gir som resultat firmaets navn.
FADRESSE	:	Gir som resultat firmaets gateadresse.
FTELEFON	:	Gir som resultat firmaets telefon-nummer.
FSTED	:	Gir som resultat firmaets stedsadresse.
FTELEX	:	Gir som resultat firmaets telexadresse.

Definisjon:

```
^MD/HODE/^<^NF;^BT=+;^JM=L;^FM=NO;^HD,,-;^HD,,-;
^HU,,-;^aw,000    000 0000000    ;
^HU,,-;^aw,0000   000 00000000   ; ^1/^<^FIRMA;^>;
^HU,,-;^aw,00000 000 000000000 ;
^HU,,-;^aw,000000000 000    000 ;
^HU,,-;^aw,000000000 000    000 ; ^2/^<^FADRESSE;^BL=-1;^JM=R;^FTELEFON;^>; ^JM=L
^HU,,-;^aw,000 00000 000000000 ;
^HU,,-;^aw,000   0000 00000000   ;
^HU,,-;^aw,000    000 0000000    ; ^3/^<^FSTED;^BL=-1;^JM=R;^FTELEX;^>;
^JM=L;^HD,,-;^HD,,-;^HD,,-;^HD,,-;^HD,,-;^HD,,-;^OF;^>;
```

Som man kan se er denne makroen relativ ukomplisert.

Det er her en ting som bør bemerkes spesielt:

> Når man benytter kombinasjoner av *HD-* og *HU-direktiver* uten automatisk retur (parameterverdien '-'), MÅ det forekomme like mange *HU* som *HD* på én og samme tekstside. Hvis ikke vil sidehaleteksten bli forskjøvet.

Bruken av *HU* og *HD* i denne makroen gjør at linjeavstanden for selve logoen blir halvert i forhold til aktuell linjeavstand for den øvrige teksten.

Makroen innledes med å lagre unna alle formatparametre *[^NF;]*. Uthevet skriftmodus slås på *[^BT=+;]*. Justering og fyllingsmodus defineres *[^JM=L;^FM=NO;]*.

Deretter følger så to *HD* uten retur *[^HD,,-;^HD,,-;]*, dette betyr at det gjøres to halve linjeskift uten at papiret går tilbake til utgangspunktet. Hvor mange av *HD'ene* som skal komme før og etter logoen er egentlig likegyldig for resultatet av selve logoen, men det har betydning for logoens plassering i forhold til den øvrige teksten.

Vi har nå kommet fram til første tekstlinje i selve logoen, som innledes med et halvt linjeskift opp uten retur *[^HU,,-;]*. Selve tekstdelen skrives ved hjelp av *AW-direktivet* (absolutt-ord) *[^aw ooo ooo oooooooo].;* Dette gjør at alle tegnene blir betraktet som ett ord, og de blir derfor ikke innbyrdes justert.

Neste linje i logoen lages på samme måte, men denne etterfølges av ekstra tekst, parameter 1 til makroen med standardverdi firmaets navn: *[^1/^<^FIRMA;^>;]*.

Linje 3 og 4 lages på samme måte som linje 1.

Linje 5 i logoen etterfølges også av ekstra tekst, parameter 2 til makroen, der firmaets gateadresse og telefonnummer er standardverdi, henholdsvis venstre- og høyre-justert på linjen til høyre for logoteksten: *[^2/^<^FADRESSE;^BL=-1;^JM=R;^FTELEFON;^>;]*

Deretter settes justeringsmodus tilbake til venstrejustering *[^JM=L;]*.

Linje 6 og 7 lages på samme måte som linje 1.

Linje 8 i logoen etterfølges også av ekstra tekst, parameter 3 til makroen, der firmaets stedsadresse og telexadresse er standardverdi, henholdsvis venstre- og høyre-justert på linjen til høyre for logoteksten: *[^3/^<^FSTED;^BL=-1;^JM=R;^FTELEX;^>;]*

Deretter settes justeringsmodus tilbake til venstrejustering *[^JM=L;]*.

For å oppveie de 8 *HU-direktivene* som er brukt til å lage logoteksten, følger så 6 *HD-kall* (pluss de 2 i innledningen blir dette også 8).

Makroen avsluttes så ved at alle formatparametre settes tilbake til sine opprinnelige verdier *[^OF;]*.

Dersom man ønsker å forandre på denne logoen, noe som sikkert er aktuelt for de fleste, må man være oppmerksom på at det er plass til et ganske begrenset antall bokstaver av denne størrelsen på en "linje". Spesielt dersom man også ønsker å ha med firmanavn osv., slik som i denne makroen.

10.11 MAKROEN "NOTAT"

Denne makroen er i klassen dokumentmakroer, og brukes for å skrive dokumenter av type *NOTAT*.

Makroen kan kalles med opptil 6 parametre:

1. Til , hvem notatet er rettet til.
2. Kopi til (opsjon), hvem som bør motta kopi av notatet.
3. Fra , hvem notatet er sendt fra.
4. Sak , hva notatet omhandler.
5. Deres ref (opsjon),
6. Vår ref (opsjon).

Definisjon:

```
^MD/NOTAT/^<^DX;^PH=0;^TB=0;
  ^HODE/ / /^<_^FIRMA;       N O T A T_^>;^IR-SCH,NOTAT;
  ^MD=SAK=^4=^<^OP=Sak: ;^>;;
  ^JM=L;^FM=F;^SIP,,10,0;
  ^IP=Til:       ;^IR-TON,1;^1=^<^OP=Til:       ;^>;^IR-TOFF;^BL;
  ^IF|''^2/*-*;'<>'*-*'|^<^IP=Kopi til: ;^IR-TON,2;^>^2;^<^IR-TOFF;^>;^BL;
  ^RM=31;
  ^IP=Fra:       ;^IR-TON,3;^3=^<^OP=Fra:       ;^>;^IR-TOFF;^BL=-1;
  ^RM=0;^LM=+^AR.^$PW;*2/5;;
  ^IF|''^5/*-*;'<>'*-*'|^<^IP=Deres ref: ;^IR-TON,4;^>^5;^<^IR-TOFF;^>;^BL;
  ^IF|''^6/*-*;'<>'*-*'|^<^IP=Vår ref:   ;^IR-TON,5;^>^6;^<^IR-TOFF;^>;^BL=1;
  ^LM=-^AR.^$PW;*2/5;;
  ^IP=Dato:      ;^IR-TON,6;^FDATO;^IR-TOFF;^BL=1;
  ^IP=Sak:       ;^IR-TON,7;_^SAK;_^IR-TOFF;
  ^EIP;^H1=_^SAK;_;^FM=C;^SI=0;^PI=0;^JM=S;^TL=forts....;
  ^PH=1;^TB=4;^BL=2;^EOD-STR=^<^TL= ;^>;^IR-TON,8;^>;
```

Som du ser inneholder denne makroen en del direktiv-kall av typen *^IR-*. Dette er direktiver som forbereder en konvertering av dokumentet for lagring ved hjelp av *NOTIS-IR*. Disse direktivene vil ikke bli nærmere forklart her, da bruken av dem er avhengig av skjema-definering i NOTIS-IR. De vil også bli utelatt fra forklaringen slik at de ikke skal gjøre den uoversiktlig.

Denne makroen er relativt ukomplisert og rett frem, og vi forklarer den derfor trinn for trinn uten å dele den opp i seksjoner.

Den innledes med å spesifisere formatering for dobbeltsidig kopiering: *[^DX;]*. Deretter slås sidehode-funksjonen av, og øvre kant settes lik 0: *[^PH=0;^TB=0;]*. Dette gjøres for at første side ikke skal ha noe sidehode i vanlig forstand, men et notat-hode.

Det utføres så et kall til logo-makroen, hvor parameter 1 og 2 settes til blank, mens parameter 3 blir firmaets navn samt teksten *N O T A T*, understreket:
[^HODE/ / /^<_^FIRMA; N O T A T_^>;].

Deretter defineres hjelpemakroen SAK til å inneholde parameter 4. Dersom parameter 4 ikke er spesifisert, vil verdien bli forespurt på terminalen under formateringen: *[^MD=SAK=^4=^<^OP=Sak: ;^>;;]*.

Justering og fyllingsmodus defineres, og det forberedes for innrykkede avsnitt: *[^JM=L;^FM=F;^SIP,,10,0;]*.

Ledeteksten "Til: " med tilhørende tekst fra parameter 1 skrives ut som innrykket avsnitt. Dersom parameter 1 ikke er spesifisert, vil verdien bli forespurt på terminalen under formateringen:
[^IP=Til: ;^1=^<^OP=Til: ;^>;^BL;].

Det testes så på hvorvidt parameter 2 er spesifisert (denne type testing er forklart tidligere). Dersom denne parameteren er spesifiert, skrives ledeteksten "Kopi til: ", samt parameterteksten som innrykket avsnitt. Dersom parameter 2 ikke er spesifisert, vil heller ikke ledeteksten bli skrevet:
[^IF|''^2/-*;'<>'*-*'|^<^IP=Kopi til: ;^>^2;;*
].

Høyre marg økes, for å sette av plass til referansetekstene:
[^RM=31;]. Deretter skrives ledeteksten "Fra: " etterfulgt av parameter 3. Dersom denne parameteren ikke er spesifiert, vil den bli forespurt på terminalen under formateringen:
[^IP=Fra: ;^3=^<^OP=Fra: ;^>;].

Dette linjeinnholdet skrives så ut, UTEN linjeskift: *[^BL=-1;]*. Dette betyr at etterfølgende tekst vil komme på samme linje.

Høyre marg settes tilbake til 0, og venstre marg beregnes som en funksjon av sidebredden (denne kan selvfølgelig settes som en konstant): *[^RM=0;^LM=+^AR.^$PW;*2/5;;]*.

"Deres ref" og "Vår ref" skrives så ut på hver sin linje i henhold til de nye margene, *dersom de er spesifisert*. Dette utføres på samme måte som for parameter 2:

```
[^IF|''^5/*-*;'<>'*-*'|^<^IP=Deres      ref:       ;^>^5;;^BL;
^IF|''^6/*-*;'<>'*-*'|^<^IP=Vår ref:   ;^>^6;;^BL=1;].
```

Venstre marg settes så tilbake til opprinnelig verdi:
*[^LM=-^AR.^$PW;*2/5;;].*

Ledeteksten "Dato: ", etterfulgt av datoen da dokumentet var sist oppdatert, skrives som et innrykket avsnitt, etterfulgt av en blank linje: *[^IP=Dato: ;^FDATO;^BL=1;].*

Deretter skrives ledeteksten "Sak: " etterfulgt av tilhørende tekst understreket, som et innrykket avsnitt:
[^IP=Sak: ;_^SAK;_].

Seksvensen av innrykkede avsnitt avsluttes: *[^EIP;].*

Hode-1 defineres til å inneholde saksteksten understreket:
[^H1=_^SAK;_;].

Fyllings og justeringsmodus, samt seksjons- og avsnittsinnrykk, defineres: *[^FM=C;^SI=0;^PI=0;^JM=S;].*

Sidehalen defineres til å inneholde teksten "forts...":
[^TL=forts....;].

Sidehode-funksjonen slås på igjen, og øvre kant settes tilbake til standardverdi: *[^PH=1;^TB=4;].*

NOTAT-hodet avsluttes med to blanke linjer: *[^BL=2;].*

Makroen avsluttes så ved at *EOD-STR* (End-Of-Document STRing) defineres til å sette blank sidehale. Dette betyr at alle sider med unntak av den siste vil få "forts..." i sidehalen: *[^EOD-STR=^<^TL= ;^>;].*

Dersom du tar for deg en dokumentmakro på denne måten og går igjennom den trinn for trinn, er de fleste ikke så veldig vanskelige å modifisere. Spesielt ikke dersom du kun ønsker å modifisere ledetekster, e.l.

LYKKE TIL!

E K S E M P E L O V E R S I K T

Eksempel		Side
1	Definering og kalling av enkel brukermakro	2
2	Eksempel på sekvensen '^'	11
3	Nivå-begrepet ved kalling av makro	12
4	Nivå-begrepet ved kalling av makro	12
5	Makrodefinering uten sitattegn	14
6	Makrodefinering med sitattegn	14
7	Ekspansjon av makro med variabel stamme	15
8	Bruk av sitattegn på deler av makro-stammen	16
9	Definering av parameter i stammen	16
10	Forskjellig tallrepresentasjon av system-makro	19
11	Bruk av system-makroen $SN	19
12	Definering av brukermakro med konstant stamme	22
13	Definering av brukermakro med makro-kall i stammen	22
14	Definering av brukermakro med variabel del i stammen	23
15	Definering av brukermakro med parameterinkludering	24
16	Parameterinkludering med standardverdier	25
17	Inkludering av flere parametre	26
18	Forskjellige måter å inkludere parametre	27
19	Inkludering av parameter uten ekspansjon	28
20	Redefinering av makroer	29
21	Bruk av referansemakro	38
22	Definisjon av integer-makro	42
23	Definisjon av integer-makro med aritmetisk uttrykk	42
24	Enkel trigger-makro	44
25	Køing av trigger-makroer	44

Eksempel Side

26 Triggerens prioritet 45

27 Trigger-makroens prioritet 45

28 Bruk av AR-direktivet 55

29 Bruk av IF-direktivet 58

30 Bruk av IF-direktivet 58

31 Strengsammenligning med IF-direktivet 59

32 Bruk av eksempel-makroen 68

Stikkord

AR-direktivet . . . . . 54.
aritmetiske uttrykk . . . . . 54.
balanse . . . . . 13, 16.
bruker-makro . . . . . 4, 5, 20.
CM clear-makro . . . . . 33.
definisjonstidspunkt . . . . . 22.
direktiv-slutt . . . . . 11.
direktiv-start . . . . . 11.
direktiver spesielle . . . . . 53.
DISABLE . . . . . 47.
DM dump-makro . . . . . 34.
eksempel-makro . . . . . 68.
ekspandering . . . . . 1.
ENABLE . . . . . 47.
EOD-STR . . . . . 51.
EOF-STR . . . . . 51.
FIX fikser-makro . . . . . 35.
formatparametre . . . . . 53.
henvisninger . . . . . 8.
hjelpe-makro . . . . . 77.
IF-direktivet . . . . . 56.
IM integermakro-define . . . . . 41.
integer-makro . . . . . 4, 6, 41.
kall-nivåer . . . . . 12.
kalling . . . . . 1.
LIB-CONT . . . . . 60, 64.
MA makro-append . . . . . 30.
makro . . . . . 1, 3.
  bm . . . . . 68.
  bold . . . . . 66.
  bruker . . . . . 4, 5, 20.
  CE . . . . . 69.
  CP . . . . . 70.
  element . . . . . 12.
  FIG-CP . . . . . 73.
  fiksering . . . . . 35.
  HODE . . . . . 93.
  integer . . . . . 4, 6, 41.
  NOTAT . . . . . 96.
  redefinering . . . . . 36.
  referanse . . . . . 4, 8, 37.
  SIP . . . . . 82.
  skjerm . . . . . 34.
  sletting . . . . . 29.
  spre . . . . . 71.
  system . . . . . 4, 7, 17.
  trigger . . . . . 4, 9, 43.
  uthev . . . . . 67.
  verdi . . . . . 6.
makro-definisjon . . . . . 2, 3, 21.
makro-ekspansjon . . . . . 3.
makro-kall . . . . . 2, 3, 11.

makro-navn . . . . . 5.
makro-stamme . . . . . 5.
MD makro-define . . . . . 21.
MI makro-insert . . . . . 31.
MK makro-kill . . . . . 29.
MR makro-remove . . . . . 32.
navn . . . . . 5.
NF-direktivet . . . . . 53.
nivå . . . . . 3, 12, 16.
NREDEF set-no-redefine . . . . . 36.
OF-direktivet . . . . . 53.
PAR-STR avsnitts-streng . . . . . 51.
parameter . . . . . 3-5, 11, 16, 24.
  standardverdi . . . . . 25, 27.
  tom . . . . . 25.
parameter-inkludering . . . . . 24.
parameterinkludering PM . . . . . 27.
PH-STR sidehode-streng . . . . . 50.
plassbehov . . . . . 62.
PM parameterinkludering . . . . . 27.
PRIOR . . . . . 47.
prioritet . . . . . 44.
RD reference-define . . . . . 37.
redefinering . . . . . 36.
referanse-makro . . . . . 4, 8, 37.
ressursforbruk . . . . . 62.
retning . . . . . 32.
single-trigger . . . . . 9, 48.
sitattegn . . . . . 4, 14.
skilletegn . . . . . 11.
stamme . . . . . 5.
  fjerning . . . . . 32, 33.
  konstant . . . . . 22.
  makro-kall . . . . . 22.
  makro-kall-sitattegn . . . . . 23.
  tillegg . . . . . 30, 31.
stamme-element . . . . . 30.
størrelse . . . . . 8, 37.
syntaks . . . . . 11.
system-makro . . . . . 4, 7, 17.
TL-STR sidehale-streng . . . . . 50.
TM triggermakro . . . . . 43.
tom parameter . . . . . 25.
TP-STR tittelside-streng . . . . . 50.
trigger single . . . . . 9, 48.
trigger-makro . . . . . 4, 9, 43.
trigger-streng . . . . . 10, 49.
type . . . . . 6, 7.
UND-STR understrekings-streng . . . . . 52.
verdi . . . . . 6, 18, 41.
vilkår . . . . . 9, 43.

********** SEND US YOUR COMMENTS!!! **********

Are you frustrated because of unclear information in this manual? Do you have trouble finding things? Why don't you join the Reader's Club and send us a note? You will receive a membership card — and an answer to your comments.

Please let us know if you

- find errors
- cannot understand information
- cannot find information
- find needless information

Do you think we could improve the manual by rearranging the contents? You could also tell us if you like the manual!

********** HELP YOURSELF BY HELPING US!! **********

Manual name: NOTIS TF – Makro Veiledning

Manual number: ND-63.010.01

What problems do you have? (use extra pages if needed) _____

_____

_____

_____

_____

Do you have suggestions for improving this manual ? _____

_____

_____

_____

_____

_____

Your name: _____ Date: _____

Company: _____ Position: _____

Address: _____

_____

What are you using this manual for ? _____

_____

NOTE!
This form is primarily for documentation errors. Software and system errors should be reported on Customer System Reports.

Send to:
Norsk Data A.S
Documentation Department
P.O. Box 25, Bogerud
0621 Oslo 6, Norway

Norsk Data's answer will be found on reverse side

Answer from Norsk Data _____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Answered by _____ Date _____

Norsk Data A.S
Documentation Department
P.O. Box 25, Bogerud
Oslo 6, Norway