

context

Beginnen met ConT_EXt

Berend de Boer

Volgens Hans zou ik de eerste ConT_EXt gebruiker buiten Pragma zijn. Ik heb dat eens nagekeken: ik heb hier nog een context format uit 1996. Dat zou dan de eerste context.fmt zijn die Pragma heeft verlaten, een collectors item :-).

Ik gebruik T_EXsinds 1989 of 1990. Ik geloofde toen nog dat WordPerfect 5.1 het eind van alle tekstverwerkers was. Maar laten we dat maar snel vergeten, het helpt me in elk geval begrip op te brengen voor Word gebruikers...

Voor veel documenten maakte ik custom layouts. Dat hield dus in L_AT_EX sources aanpassen. L_AT_EX is slecht aanpasbaar of instelbaar, dus meestal kwam dat neer op het kopiëren van veel macro's uit latex.fmt en een aantal niveaus vervolgens hacken. Bij de volgende versie van L_AT_EX begon het spel weer opnieuw.

Toen ik Hans ontmoette beweerde hij dat dat allemaal veel eenvoudiger kon. Ik ben op een avond bij Pragma geweest met een layout die mij bloed, zweet en tranen had gekost. Voor een shareware pakket had ik de layout van Borland's handleidingen gecomplementeerd. Ik was vrij sceptisch over hoe Hans zou laten zien dat dit in een handomdraai nagemaakt kon worden. Verbazen zal ik de ConT_EXt gebruikers niet: Hans kon dat inderdaad heel erg snel.

Op die avond was ik om. Ik kreeg een ConT_EXt handleiding, format files en een speciale versie van emTeX: HugeTex. Thuis werkte alles vrijwel meteen. Toch heeft het nog vrij lang geduurd voordat ik vrijwel uitsluitend ConT_EXt gebruikte. Ik stapte in die tijd ook over op Unix als mijn primaire platform. ConT_EXt 'draaide' nog niet onder Unix omdat Hans de sources nog niet had vrijgegeven, alleen de .fmt bestanden. En het valt niet mee als je erg gewend bent aan bepaalde macro's om dan opeens alles te doen met geheel andere macro's. Ik was dan ook in het begin veel tijd kwijt met het opzoeken van hoe stel ik dit in en hoe stel ik dat in.

Ik denk dat dit laatste dan ook de grootste hobbel is die ConT_EXt zal moeten nemen. Te veel mensen hebben met L_AT_EX leren leven. Met dit artikel probeer ik daar een beetje verandering in te brengen door te laten zien hoe je met ConT_EXt layouts kunt instellen.

De volgende zaken komen aanbod:

1. Hoe stel je ConT_EXt standaard in dat je PDF uitvoer krijgt.
2. Hoe wijzig je het papierformaat.
3. Hoe stel je een basis font in.
4. Hoe wijzig je de hoofdstuk en paragraaf kopjes.
5. Waar komt het paginanummer.
6. Hoe krijg ik eigen hoofd- en voetteksten.
7. Kan ik ook nog ergens mijn tekst intypen asjeblijft?
8. Hoe voeg je een plaatje toe.
9. Hoe genereer je een inhoudsopgave.
10. Hoe genereer je een index.

Ik richt me in dit verhaal vooral op gebruikers die PDF uitvoer willen.

Verder ga ik uit van de allerlaatste ConT_EXt versie. Voorgaande versies deden dingen anders.

PDF uitvoer

De eerste beslissing die ik altijd neem is PDF output te gebruiken. De documenten die ik maak zijn vrijwel uitsluitend bestemd voor een publiek dat niet over een dviewer beschikt. PDF is ideaal als platform onafhankelijke uitvoer.

Als eerste regel in een document neem ik dus dit op:

```
\steluitvoerin[pdftex]
```

Omdat ik dit voor alle documenten wil, heb ik dit commando in een speciaal bestand geplaatst, namelijk cont-sys.tex. Dit bestand leest ConT_EXt bij het opstarten.

Paginaformaat instellen

Het paginaformaat is het volgende commando. Dat is in te stellen met:

```
\stelpapierformaatin[A4][A4]
```

De eerste parameter legt het formaat vast waarmee ConT_EXt werkt bij het typesetten van de tekst. De tweede parameter legt het formaat vast wat uit de printer rolt. Veel meer keus dan A4 is er, praktisch gezien, meestal niet.

Fonts instellen

De volgende stap is het instellen van het basisfont. Dat kan met dit commando:

```
\stelkorsin[ber,phv,ss,11pt]
```

De eerste parameter geeft aan dat we de Karl Berry encoding hanteren, volgens mij de meest gebruikelijke hantering voor postscript fonts. De aanduiding “ber” is dus zo wat verplicht als je je eerste postscript font opgeeft, daarna is deze niet meer nodig.

De tweede parameter geeft aan dat we het font Helvetica willen laden, een font zonder schreefje. Dit font is ongeveer gelijk aan Arial, wat Windows gebruikers wellicht beter kennen.

De derde parameter geeft aan ConTeXt door dat we nu gaan switchen naar dit font—zonder—schreefje, “sans serif” dus. Doordat we eerst Helvetica hebben geladen is het standaard ‘ss’ font overschreven.

De derde parameter geeft de font grootte op.

Een voordeel van een standaard postscriptfont is dat het PDF bestand kleiner blijft. Er hoeft geen extra font informatie in te worden opgenomen.

Ruimte tussen regels instellen

Als het font wat groter is als het standaard font van 10 punten, dan is het nodig om ook de regelafstand wat aan te passen. Het volgende commando zorgt er voor dat de standaard regelafstand iets kleiner wordt.

```
\stelinterliniein[regel=13pt]
```

Ruimte tussen paragrafen instellen

Meestal wil ik een lege regel tussen paragrafen. Dat kan met dit commando:

```
\stelwitruimtein[groot]
```

Instellen van hoofdstuk en paragraaf kopjes

De standaard kopjes zijn niet vet. Met het volgende commando is het mogelijk om de kopjes van de hoofdstukken, paragrafen en subparagrafen vet te zetten.

```
\stelkopin[hoofdstuk][letter=\bfc]
\stelkopin[paragraaf][letter=\bfa]
\stelkopin[subparagraaf][letter=\bf]
```

De eerste parameter verwijst naar hetgene wat je wilt instellen: hoofdstuk, paragraaf, subparagraaf, subsubparagraaf, etc.

De tweede parameter is de instelling zelf. Hier wijzig ik alleen de letter. Maar er zijn nog veel meer instelling. Met

het volgende commando plaats je bijvoorbeeld de nummers in de marge:

```
\stelkopin[hoofdstuk][letter=\bfc,variant=inmarge]
\stelkopin[paragraaf][letter=\bfa,variant=inmarge]
```

Het is ook mogelijk om de kop geheel zelf te formateren. Bijvoorbeeld het hoofdstuknummer kan als volgt geformatteerd worden:

```
\def\kophfd#1#2{{\bfc Nummer: #1, Titel: #2}
\stelkopin[hoofdstuk][commando=\kophfd]
```

Instellen van paginanummer

Het paginanummer komt standaard middenin, bovenaan te staan. Het volgende commando plaatst het paginanummer onderaan.

```
\stelnummeringin[plaats={midden,voet}]
```

Het paginanummer wordt met dit commando uitgezet, want in de volgende paragraaf worden hoofd- en voetteksten ingesteld met paginanummer.

```
\stelnummeringin[plaats=]
```

Als het paginanummer niet uitgezet wordt, komt het over de hoofd- en voetteksten heen.

Instellen van hoofd- en voetteksten

Het is eenvoudig zelf hoofd- en voetteksten in te stellen. De volgende twee commando’s definiëren de opmaak. De laatste twee commando’s maken deze opmaak actief.

```
% eigen koptekst
\def\mykopstekst{%
\ vbox{%
\ hbox to \tekstbreedte{\bfb Test\hfill Pagina}
\ par\haarlijn
}
}
% en eigen voetekst
\def\myvoetekst{%
\ vbox{%
\ haarlijn\par
\ hbox to \tekstbreedte{%
\ huidigedatum
\ hfill
{\bf Mij bedrijf}\hfill
Pagina~\paginanummer~van~\totaalaantalpaginas
}
}
}
% instellen
\stelhoofdstekstenin[\mykopstekst]
```

```
\stelvoettekstenin[\myvoettekst]
```

Voor degenen die dit niet kennen: met het commando `\hfill` is het mogelijk om dingen naar links en naar rechts te ‘duwen’. Een `\hfill` wordt net zo groot als mogelijk. Neem je twee `\hfill`’s op, dan worden ze allebei even groot. Dus de opmaak in `mykoptekst` duwt de tekst ‘Test’ naar links en de tekst ‘Pagina’ naar rechts totdat ze allebei tegen de kantlijn aanstaan, en dus niet verder geduwd worden.

Om ze tegen de linker en rechterkantlijn te krijgen wordt een horizontale box gemaakt waarbinnen dit allemaal gebeurt. Die horizontale box is net zo groot als de breedte van de tekst.

Als deze box, commando `\hbox`, niet geplaatst zou zijn, kreeg je niet het gewenste effect, want dan zou de ruimte rechts ‘beperkt’ worden door de `\par`.

De tekst zelf

Na al deze instellingen zou je de tekst zelf bijna vergeten! Tekst wordt geplaatst na het commando `\starttekst`. De tekst eindigt met `\stoptekst`.

```
\starttekst
```

```
Hier komt mijn tekst.
```

```
\stoptekst
```

Hoofdstukken en paragrafen worden als volgt gedaan:

```
\starttekst
```

```
\hoofdstuk{Mijn hoofdstuk}
```

```
\paragraaf{Mijn paragraaf}
```

```
Hier komt mijn tekst.
```

```
\stoptekst
```

Verwijzen naar paragrafen kan met het `\in` commando.

```
\starttekst
```

```
\paragraaf{Mijn eerste paragraaf}
```

```
Hier komt mijn tekst, zie ook
paragraaf~\in[par:twee].
```

```
\paragraaf[par:twee]{Mijn tweede paragraaf}
```

```
En hier nog meer.
```

```
\stoptekst
```

De paragraaf krijgt dus een extra parameter, tussen rechte haken. Tussen die rechte haken komt een volkomen wille-

keurige tekst waarnaar verwezen kan worden met het `\in` commando.

Opnemen van een plaatje

Plaatjes opnemen is heel eenvoudig geworden met de laatste ConT_EXt en pdftex versies. Neem dit commando op om een plaatje te definiëren:

```
\gebruikexternfiguur[MijnPlaatje] []
                             [type=png, schaal=600]
```

De eerste parameter is de naam van het commando waarmee je dit plaatje op kunt roepen. De tweede parameter is de naam van het bestand van dit plaatje. Als dit leeg gelaten wordt, is dit gelijk aan de eerste parameter.

Met de derde parameter kan het type opgegeven worden, en kan het plaatje eventueel geschaald worden. Dit is afhankelijk van de T_EX versie die gebruikt wordt. Met pdftex werkt alles prima.

Het plaatje kan daarna opgeroepen en geplaatst worden met:

```
\plaatsfiguur
  [hier]
  [fig:MijnPlaatje]
  {Een mooi plaatje}
  {\MijnPlaatje}
```

De eerste parameter geeft aan waar het plaatje geplaatst mag worden.

De tweede parameter is een referentie waarmee naar het plaatje verwezen kan worden, zie onder. Een referentie is een volkomen willekeurige tekst.

De derde parameter is de bijbehorende tekst.

De vierde parameter is het commando voor het plaatje zelf.

Grootte e.d. worden automatisch bepaald door ConT_EXt mits een recente T_EX gebruikt worden.

Verwijzen naar dit plaatje kan via:

```
Figuur~\in[fig:MooiPlaatje] bevat een
mooi plaatje.
```

Genereren van een inhoudsopgave

Om de inhoudsopgave te kunnen maken zijn twee slagen nodig. Eerst moet ConT_EXt het bestand een keer verwerken. Er wordt dan een bestand met de extensie `.tui` gecreëerd. Dit bestand kan met `texutil` verwerkt worden:

```
texutil --references test
```

Als ik dat intype, krijg ik de volgende output:

```
/home/berend/tmp# texutil.pl --references test
```

```
TeXUtil 6.6 - ConTeXt / PRAGMA 1992-1998
```

```

    action : processing commands, l\
            ists and registers
    input file : test.tui
    output file : test.tuo
    passed commands : 6
    register entries : 0 -> 0 entries 0 refer\
                    ences
    synonym entries : 0 -> 0 entries
    embedded files : 1
/usr/home/berend/tmp#

```

texutil is een commando dat bij de ConTeXt distributie geleverd worden.

Daarna kan de inhoudsopgave geplaatst worden door dit commando op de gewenste plek neer te zetten:

```
\volledigeinhoud
```

Voor mensen die de layout van de inhoudsopgaven willen instellen is er het `\stelsamengesteldelijstin` commando.

```

\stelsamengesteldelijstin
[inhoud]
[variant=c,
niveau=subparagraaf]

```

Genereren van een index

Een index wordt op dezelfde wijze als de inhoudsopgave gegenereert. Eerst moet ConTeXt het bestand een keer verwerkt hebben, daarna kan texutil met parameter `--references` aangeropen worden.

De inhoudsopgave wordt met het volgende commando geplaatst.

```
\volledigregister[index]
```

Om iets in de index te krijgen, zijn er twee commando's. Met `\index` kan iets in de index geplaatst worden. Met `\zieindex` kan een verwijzing binnen een index gemaakt worden.

```

\zieindex{hallo wereld}{hello world}

\starttekst
De meest bekende test--tekst is
wel \index{hello world}'hello world'
\stoptekst

```

Het commando `\zieindex` zal meestal in het instelgebied, dus voor `\starttekst` terecht komen. Het commando

`\index` wordt gebruikt *voor* de tekst waar deze naar verwijst.

Waar is meer informatie te vinden

Ik heb ConTeXt vrijwel uitsluitend geleerd met behulp van de handleiding in `ms-co-nl.pdf`. Voor de meeste instellingen vond ik deze handleiding voldoende.

Verder gebruik ik veel het tooltje `grep`. Dit tooltje kan woorden opzoeken in tekst. Wil ik bijvoorbeeld weten waar in de ConTeXt sources het commando `\stelpapierformaat`in voorkomt, dan type ik:

```
/usr/local/share/texmf/tex/context/base# grep -n\
"stelpapierformaat" *.tex
```

```

core-01a.tex:30:% \stelpapierformaatin[liggend,A\
                    4][liggend,A4]
core-01a.tex:1649:%I \stelpapierformaatin[DIN-\
                    formaat]
core-01a.tex:1661:%I \stelpapierformaatin[A5][A4]
core-01a.tex:1708:\def\dostelpapierformaatin[#1]\
                    [#2]%
core-01a.tex:1744: \stelpapierformaatin[#1][#2]%
core-01a.tex:1747:\def\stelpapierformaatin%
core-01a.tex:1748: {\dodoubleempty\dostelpapier\
                    formaatin}
...

```

Vervolgens kan ik in 1 van de opgesomde bestanden kijken of er nog iets extra's over dit commando vermeld is, wat nog niet in de handleiding staat. De sources lezen doe ik nauwelijks. TeX sources zijn vrij onleesbaar wat mij betreft, maar je krijgt wel de indruk bij de ConTeXt sources dat ze leesbaar zijn, een heel verschil als je `latex.ltx` bestudeert hebt.

Conclusie

Ik hoop dat dit artikel mensen heeft aangezet om ConTeXt eens uit te proberen. Ik geef toe dat TeX pakketten voor de gemiddelijke gebruiker niet makkelijk te installeren zijn. Maar voor wie de luxe heeft een systeembeheerder te hebben: vraag hem om ConTeXt ook beschikbaar te maken. Dit kost hem waarschijnlijk weinig moeite.

Voor mensen die verder vragen en of opmerkingen hebben over ConTeXt: welkom bij de ConTeXt mailling list. Stuur een mailtje naar `majordomo@ntg.nl` met als inhoud "subscribe ntg-context".