

Siep Kroonenberg  
Faculteit der Economische Wetenschappen  
Rijksuniversiteit Groningen  
n.s.kroonenberg@eco.rug.nl

#### abstract

Introductie van en toelichting bij het nieuwe NTG-logo

#### keywords

logo, Encapsulated PostScript



In het vorige nummer van de MAPS heeft u kunnen lezen over de ontwerpwedstrijd voor een nieuw logo. En hier is dan het nieuw logo. U ziet dat in geen enkele versie de tekst 'Nederlandstalige T<sub>E</sub>X gebruikersgroep' voluit voorkomt. Het leek me beter om per toepassing daar een keus voor te maken.

Ik heb met een aantal ideeën gespeeld:

1. iets dat lijkt op het T<sub>E</sub>X logo, met de 'T' naar boven gewipt
2. een monogram met meer dan 1 font: een strakke, geometrische 'T', en 'N' en 'G' uit een minder schematisch font.
3. elke letter in zijn eigen rechthoek plaatsen; dit als verwijzing naar de manier waarop T<sub>E</sub>X tekst zet. Hier is het uitkijken geblazen dat het logo er niet warrig uit gaat zien.

Deze ideeën leidden tot een aantal logo's; een combinatie van idee 2 en 3, met font cmb10:



Het is niet strikt noodzakelijk om T<sub>E</sub>X-fonts te gebruiken; het volgende logo, een implementatie van idee 1, gebruikt

## Het nieuwe NTG logo

b.v. Bitstream's versie van Kabel. Een citaat uit de Adobe Type On Call CD:

*Designed by Rudolf Koch and released in 1927 by the Klingspor foundry in Germany, Kabel is named in honor of the laying of the first trans-Atlantic telephone cable.*



De dwarsbalk van de 'T' is iets naar links verlengd. Een tweede versie is wit uitgespaard tegen een donker vlak. Hierbij is de afstand tussen de letters een fractie vergroot om optische effecten te compenseren.

Het winnende logo combineert idee 1 en 3. Ik was gecharmeerd door het stijlcontrast tussen de zeer simpele ronde lettervormen en de dunne strakke rechthoekjes. Omdat echter die rechthoekjes bij verkleining niet meer goed tot hun recht komen, heb ik ook een versie zonder rechthoekjes gemaakt:



Het font is Formal Script 421, Bitstream's versie van Ondine. Wederom een citaat uit Type On Call:

*Released by the French type foundry Deberny & Peignot in 1954, this bold upright script was an early design of Adrian Frutiger and the only script face he ever created.*

Toch eens kijken of een soortgelijk effect niet met een T<sub>E</sub>X font te bereiken valt. Euler (eusb10) ligt het dichtst in de buurt:



Maar dit wil echt niet.

Mijn eigen voorkeuren waren het Kabel- en het Ondine logo, en het bestuur koos voor het laatste.

De kleur van het logo – als kleur werd gebruikt, b.v. op voorbedrukt briefpapier en voor de web-pagina's – zou identiek moeten zijn aan de MAPS-kleur. Omdat beeldschermen geen heldere blauw-groene kleuren kunnen produceren, werd de web-versie een veel stemmiger blauw-groen. Toen men die kleur zag werd besloten die ook voor gedrukte toepassingen te gebruiken. Voor de prepress-kenners: het Pantone-nummer is 549.

Op het moment van schrijven heeft het bestuur zich nog

niet uitgesproken over een brief-stijl, maar als u dit leest heeft u waarschijnlijk al kunnen zien wat daar uit is gekomen.

Bij een logo kunnen letters het beste naar contouren worden omgezet. Daardoor hoeft het font niet beschikbaar te zijn bij het afdrucken van het logo. Veel tekenprogramma's kunnen dit en een logo in eps- (Encapsulated PostScript) formaat opslaan, en dat is wat ik heb gedaan. Voor niet-PostScript printers zal het logo voor een aantal resoluties worden omgezet naar bitmapped formaten.

## Why `\expandafter` is sometimes needed by common users too.

1. When dealing with textual input, that is reading lines of text,  $\TeX$  is nearly always expanding whatever it encounters. There are two primitives that can influence this process: `\noexpand` and `\expandafter`. The first primitive will probably never surface in user input, but the latter may! Let's give an example. In most cases one will call for a new chapter by saying something like:

```
\Chapter{This Or That}
```

or

```
\Chapter{\ThisOrThat}
```

When `\ThisOrThat` has only one meaning and is never changed, this goes all right. The title is typeset and when called upon, it appears in the table of contents too.

2. Macro packages use auxiliary files to save entries of for instance tables of contents and indexes. When writing a chapter title to such a file, we can:

- expand the macros that are part of the title
- selectively expand those macros
- copy the title as is without expansion

The first alternative is the most simple. However, certain macros can expand into long sequences of tokens, that in the worst case are only partially expanded. Font switches are an example of such fragile commands.

A solution to this problem is partial expansion. This can be accomplished by preventing certain macros from expanding. This can be done quite easily at the macro programming level by preceding such commands by a `\noexpand` when the moment is there. But which macros need such precautions and which ones don't? And how

is the user supposed to know this? Even worse, we expect users to explicitly prohibit expansion for their own macros if needed, and in practice they end up with weird sequences like:

```
\Chapter{\dontexpand\ThisOrThat}
```

This leaves option three as the most save one. This method has one disadvantage. By copying the title verbatim, we got problems when `\ThisOrThat` is used in more chapter titles with different meanings. In such cases we have to expand `\ThisOrThat` on forehand:

```
\expandafter\Chapter\expandafter{\ThisOrThat}
```

The first `\expandafter` reaches over `\Chapter` and expands the next token, `\expandafter`. This second `\expandafter` reaches over the `{` and expands `\ThisOrThat`.

When we have a bit longer title, like the next one, we end up with a lot of `\expandafter`'s:

```
\Chapter{About \ThisOrThat}
```

Out of convenience, we prefer something like:

```
\expanded{\Chapter{About \ThisOrThat}}
```

Here `\expanded` is a macro defined as:

```
\def\expanded#1%
{\edef\expandedsequence{\noexpand#1}%
\expandedsequence}
```

Hans Hagen