

# Printing labels with ConT<sub>E</sub>Xt

## Abstract

Sometimes one needs to print a single label which will be glued onto a package, a large envelope or for the identification of a box. In certain situations one wants to produce a series of identical labels or one needs to typeset whole databases of addresses. ConT<sub>E</sub>Xt offers the possibility of using the XY-arranging procedure to print on each of the labels being present on a sheet. Here a possible approach is presented for labels of the size 105 × 42.3mm i.e. (14 labels on a A<sub>4</sub>). It is shown how to print a single label but also how to get multiple copies of the same content and how to prepare sheets of labels containing the addresses of a database.

## Keywords

Maps, Context, layer, label, XY-arrangement

## Introduction

Like the address-printing on an envelope, as described in an other article in this MAPS, printing labels from a multi-label sheet is not easy in the beginning. The issue often is, that only a single label is needed. The goal is to be able to use all of the labels on a given sheet one after the other.

When creating a tool to use all the labels, the sheet must be sent through the (laser-)printer as many times as there are labels on the sheet. Due to the fact that the carrier-sheet of the labels is quite thin, it is necessary to use the labels from bottom up.

In the following article a possible setup is given. The solution makes use of the XY-arrangement and layers.

Those who want to prepare bulk mailings in The Netherlands should also think of using the KIX-code (klantenindexcode, customer index code). The Dutch mail service TNT provides a barcode-font. With this font it is possible to print this code directly on the label.

## Guidelines as provided by TNT

The Dutch Mail Service TNT provides guidelines for making a label. This information is also applicable to the printing of addresses on envelopes (See the other article in this MAPS).

In general one uses three to a maximum of six lines per address. The last line consists of the ZIP-code and the place, which is always printed in uppercase

characters. The before last line contains either the street name + house number and possibly an extension to the house number or the P.O. box + number. Before these two lines a 'to the attention of' line can be added. The other lines will contain other information concerning the address of the receiver.

In The Netherlands, one has the opportunity to add a so called customer index code KIX (klantenindexcode) in the form of a barcode. This code is unique for each address. The barcode is composed of the ZIP-code + house number (or P.O. box + number) + separator character 'X' + house number extension. The KIX-code must be printed at 10pt.

Print the KIX-code at the top of the address or as an additional line beneath the address. Always keep a minimal distance of 2mm but no more than 15mm from the last line of the address. KIX-codes should not be printed with matrix-printers due to the printout quality of these printers.

For sending mail to foreign countries the name of the receiver's country is marked down in uppercase characters as the last line of the address. Do not use KIX-codes for mail to a foreign country.

The guidelines advise you to use sans-serif fonts in the address whenever possible. Italic, script fonts, gothic fonts, matrix-characters, condensed and expanded fonts are advised against because they negatively influence the automatic read results. The character size should be no less than 7pt, nor should it exceed 17pt. It is advised to use uppercase characters throughout the address if the character size is less than 10pt.

There are even more guidelines on how to setup the alignment, the interline space and the use of spaces between words and underlining. If one lets T<sub>E</sub>X typeset the text, those aspects should be within the given rules.

## Return address

In The Netherlands one can place the return-address on top of the receiver's address. It is important that this is a single line only and it must be separated from the receiver's address by a thick rule of at least 1.2mm. The white space between the rule and the first line of the receiver's address should be 5mm.

## Installing the KIX-font for MKIV

Download the KIX-font (ttf format). Unpack the zip-file and copy the font into a font-directory, where you prepare a new foundry map e.g. “TNT” in the ttf-directory.

There are different options to make the new font known to ConT<sub>E</sub>Xt. Along the method for other fonts one can write a type-script-file containing the following lines:

```
\starttypescript [sans] [kix]
  \definefontsynonym
    [KIX-Roman]
    [file:kixbrg] [features=default]
\stoptypescript

\starttypescript [sans] [kix] [name]
  \definefontsynonym
    [Sans] [KIX-Roman] [features=default]
\stoptypescript

\starttypescript [KIX]
  \definetypface
    [KIX] [ss] [sans] [kix] [default]
\stoptypescript
```

Save the type-script-file in the user directory of ConT<sub>E</sub>Xt as type-TNT-KIX.tex. Run `luatools --generate`.

Now one can add the KIX-font in the preamble of the working file:

```
\usetypescriptfile[type-TNT-KIX]
\usetypescript [KIX]
```

For typing the actual KIX-code a small macro is defined as follows:

```
\def\KIX#1%
  {\switchtobodyfont [KIX,ss,10pt] #1}
```

Because the KIX-font does not have any different styles and one uses it invariably in a single way, there is also a shorter way to tell ConT<sub>E</sub>Xt how to use this font.

First a font synonym is created, where a symbolic name is linked to the font-file-name.

```
\definefontsynonym
  [KIX] [file:kixbrg] [features=default]
```

Again we define a macro for typesetting the KIX-code

```
\def\KIX
  {\groupedcommand
    {\definedfont [KIX at 10pt]}{}}
```

## The contents of the label

The contents will consist of two elements. First, at the top of the label, the return-address is typeset with a thick black rule underneath. For this purpose one can use a buffer.

```
\startbuffer [Returnaddress]
  \framedtext
    [frame=off,bottomframe=on,
     rulethickness=2pt,
     offset=5pt]%
    {Willi Egger,
     Maasstraat 2,
     5836 BB~{\sc Sambeek}}
\stopbuffer
```

The receiver's address is also placed in a buffer.

```
\startbuffer [Receiver]
  \framedtext{%
    \startlines
      NTG-secretary
      Maasstraat 2
      5836 BB~{\sc Sambeek}
    \stoptlines}
\stopbuffer
```

```
\startbuffer [ReceiverKIX]
  \framedtext{%
    \startlines
      NTG-secretary
      Maasstraat 2
      5836 BB~{\sc Sambeek}
      \KIX{5836BB2}
    \stoptlines}
\stopbuffer
```

```
\startbuffer [ReceiverAbroad]
  \framedtext{%
    \startlines
      NTG-secretary
      Maasstraat 2
      5836 BB~{\sc Sambeek}
      THE NETHERLANDS
    \stoptlines}
\stopbuffer
```

## Setting up the tool

The first thing one has to do is to get the precise dimensions of the labels. For this example a standard A<sub>4</sub>-label-sheet carrying 14 labels arranged in two columns is used. The sheet has no print-margins, so the labels fill the whole sheet. The labels are 105mm wide and 42.3mm high.

Next we define a paper-size with the measured dimensions of the label.

```
\definepapersize
  [Label]
  [height=42.3mm,width=105mm]
```

Now we tell ConTeXt that we will place this new paper-size on A<sub>4</sub>:

```
\setuppapersize [Label] [A4,portrait]
```

Because we will place multiple labels i.e. pages on the A<sub>4</sub> we set up the paper/sheet as follows.

```
\setuppaper
  [topspace=0mm,
   backspace=0mm,
   dx=2mm,
   dy=0mm,
   nx=2,
   ny=7,
   margin=0,
   width=210mm,
   height=297mm]
```

Now we need to instruct ConTeXt how the label/page should be set up.

```
\setuplayout
  [topspace=4mm,
   backspace=5mm,
   margin=0mm,
   width=95mm,
   height=34mm,
   header=0mm,
   footer=0mm]
```

Finally we instruct ConTeXt to use XY-arranging according to the parameters set in the `\setuppaper-block`. We will get 2 columns with 7 labels each (14 labels in total).

```
\setuparranging [XY]
```

Because the content of the buffers is in `\framedtext` we setup the behaviour of `\framedtext` with

```
\setupframedtexts
  [width=\textwidth,frame=off,offset=5pt]
```

In the following steps we set up the content of the label. One way to do this is to define a layer with the dimensions of the label.

```
\definelayer
  [Label]
  [width=\paperwidth,height=\paperheight]
```

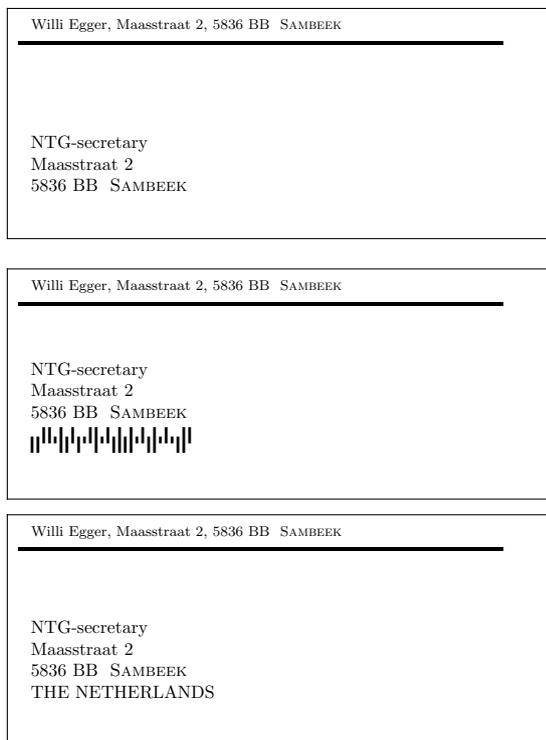
We define two variables. The first one indicates which of the labels on the sheet will be used. The second defines the total number of labels on the sheet.

```
\def\Uuselabel{8} % label(s) to be typeset
\def\Totallabels{14} % No. of labels per sheet
```

In order to get the address typeset on the correct label, we loop over the number of labels on the sheet. If the counter equals the number defined in `\Uuselabel`, ConTeXt typesets the label, otherwise it typesets nothing and moves on to the next label.

```
\dostepwiserecurse
  {1}
  {\Totallabels}
  {1}%
  {\ifnum\recurselevel=\Uuselabel
   {\setlayer
    [Label]
    [preset=lefttop,
     location={right,bottom},
     y=-5mm,x=-3.5mm]
    {\switchtobodyfont [8pt]%
     \getbuffer [Retunaddress]}}
   \setlayer
    [Etiket]
    [preset=leftbottom,
     location={right,top},
     y=.9cm,x=-3mm]
    {\switchtobodyfont [10pt]%
     \getbuffer [Receiver]}}
   \else
   {\setlayer
    [Label]
    [preset=lefttop,
     location={right,bottom}]
    {\strut}}
   \fi
   \placelayer [Label]
  \page }
```

In the future, there may be a need to define other label-sizes. Once you have more label-sizes defined, it is easier to put static information into an environment-file. All different definitions are written into a mode-paragraph. – The file we actually work with will contain the list of modes that can be enabled, the call for the environment file and definitions of the label to be used as well as the total number of labels on a sheet. The receiver's address is put in a buffer. By means



**Figure 1.** Results of the Examples of Different setups for Single Labels

of testing the active mode the respective loop-block is executed. So this file could look like this:

```

\enablemode[xxx]
\environment layout

\def\UseLabel{2}
\def\Totallabels{...}

\startbuffer[Receiver]
...
\stopbuffer

\doifmode{xxx}{%
  \dostepwiserecurse
    {1}{\Totallabels}{1}{ ... }}

```

### Typesetting multiple labels with the same content

Sometimes one would like to typeset a series of labels with the same content. What one can do, is to define the labels, which should be skipped. Define the quantity of the labels needed. By means of looping over the number of labels one can get the desired result.

```

\def\Keepempty{4}
% Skip over the first 4 labels
\def\Quantity{10}

\dorecurse{\Keepempty}{%
  \setlayer
    [Label]
    [preset=righttop,
     location={left,bottom}]
    {\strut}
  \startstandardmakeup
  \placelayer [Label]
  \stopstandardmakeup}

\dorecurse{\Quantity}{%
  \setlayer
    [Label]
    [preset=righttop,
     location={left,bottom},
     x=8mm]
    {\switchtobodyfont[10pt]%
     \getbuffer [Labelcontent]}
  \startstandardmakeup
  \placelayer [Label]
  \stopstandardmakeup}

```

### Typesetting series of labels with different content

If you have to deal with mailings, then one needs to be able to process e.g. a database with address-data. Provided that the database has the desired structure one can typeset the labels. For the labels used for the NTG-mailings, the following setup is used.

There is a layout file containing the static information. It holds a mode-paragraph defining the label-size, paper-setup, layout-setup and arranging-setup as described above.

For each member a buffer is prepared based on the members database. At the beginning of that file there is a definition of the total number of addresses contained in the database.

```

\def\Addresses{200}
...
\startbuffer[Adr52]
  \framedtext{%
    \startlines
      Dhr. W. Egger
      Maasstraat 2
      5836 BB~-\sc Sambeek}
    \stoptlines}
\stopbuffer
...

```

In the actual file, which is typeset the mode is enabled and the environment file is loaded. Then the database file is read in. Hereafter a loop over the total number of addresses is started.

```
\enablemode[ntg-Labels]
\environment layout
\input MAPS36

\dorecurse
  {\Addresses}
  {\setlayer
    [Label]
    [preset=leftbottom,
     location={right,top},
     y=-3mm,x=-3mm]
   {\switchtobodyfont[10pt]%
    \getbuffer[Adr\recurselevel]}
  \setlayer
  [Label]
  [preset=lefttop,
   location={right,bottom},
```

```
   y=.9cm,x=-3.5mm]
  {\switchtobodyfont[5pt]%
   \getbuffer[Retouraddress]}
\startstandardmakeup
\placelayer[Label]
\stopstandardmakeup }
```

## Conclusion

Label printing on a sheet containing several labels is a tricky job, due to the fact that the information must be placed very accurately. In many environments one can print labels, however it is often not possible to indicate which of the labels should be used. – With the described approach in ConT<sub>E</sub>Xt it is possible to use all the labels on a sheet. The use of loops makes it possible to typeset either multiple copies of the same information or to typeset an address-database. The tool can easily be adapted to the purpose required.

Willi Egger  
w.egger@boede.nl