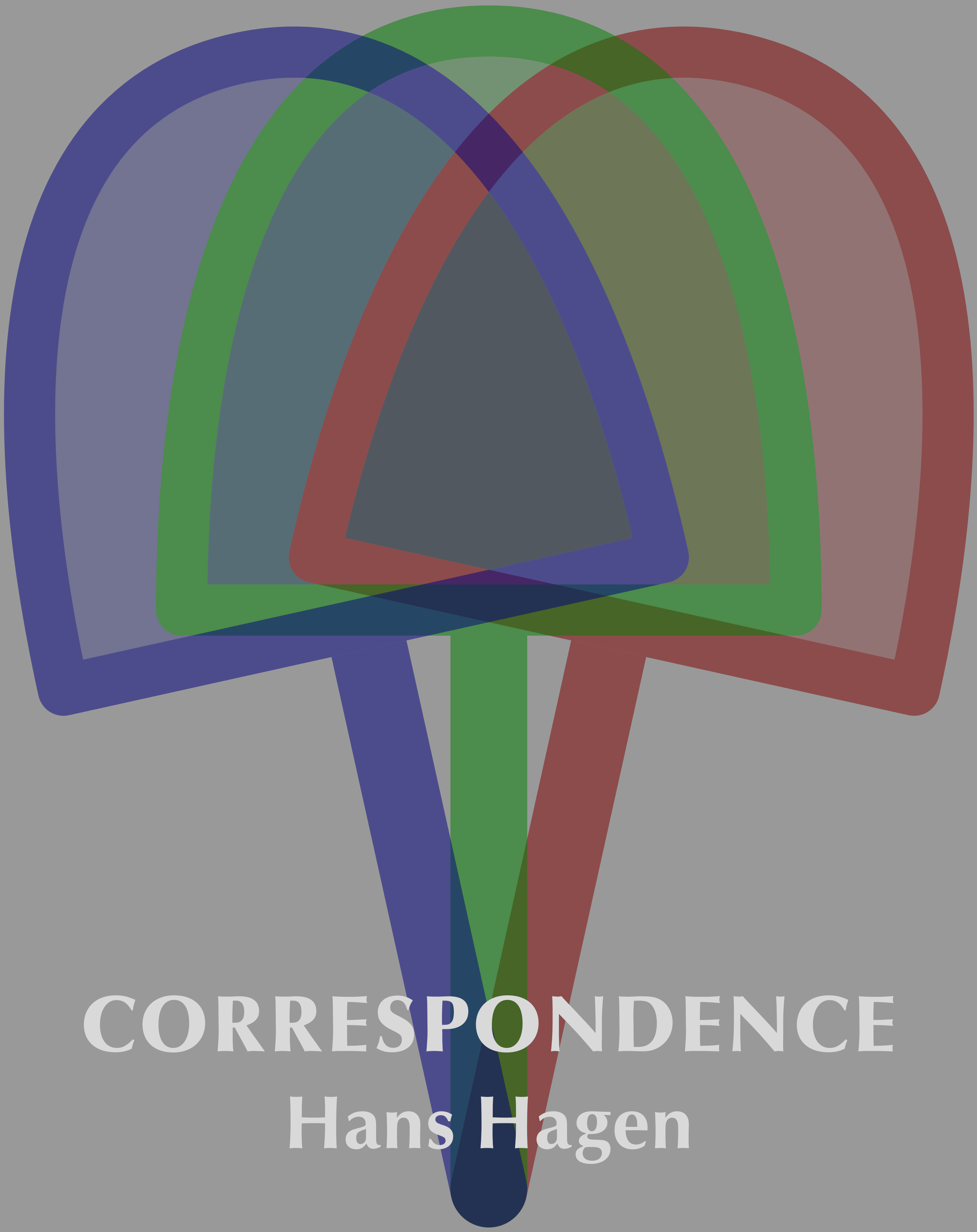


CORRESPONDENCE

Hans Hagen

1 Introduction

This manual describes how to handle correspondence with ConTEXt. The core components are a simple XML database (the successor of a TEX based format), a few styles, and a graphical user interface. It is a nice example of combined technologies.

2 The Database

One can argue about the structure of this database, but since it suits our purpose, we will not spend much words about how it ended up this way. You can happily use your own database and let it produce this kind of XML code, which is then happily accepted by ConTEXt.

As said, the structure is quite similar to the TEX counterpart that we have been using for years to typeset our letters, envelopes, mailings and alike. This is typically the kind of code that remains unchanged for years, but the advance of XML and evolution of the eXaMpLe framework triggered a rewrite.

The root element is defined as follows (we use RelaxNG as definition format).

```
<start>
  <ref name="contacts"/>
</start>

<define name="contacts">
  <element name="contacts">
    <zeroOrMore>
      <choice>
        <ref name="contacts.contact"/>
        <ref name="contacts.contactgroup"/>
        <ref name="contacts.contactfile"/>
      </choice>
    </zeroOrMore>
  </element>
</define>
```

Each contact is a collection of data fields which may come in any order.

```
<define name="contacts.contact">
  <element name="contact">
    <interleave>
      <optional> <ref name="contact.initials"/> </optional>
      <optional> <ref name="contact.formalname"/> </optional>
      <optional> <ref name="contact.informalname"/> </optional>
      <optional> <ref name="contact.title"/> </optional>
      <optional> <ref name="contact.prefix"/> </optional>
      <optional> <ref name="contact.suffix"/> </optional>
      <optional> <ref name="contact.telephone"/> </optional>
      <optional> <ref name="contact.mobiletelephone"/> </optional>
      <optional> <ref name="contact.telefax"/> </optional>
      <optional> <ref name="contact.email"/> </optional>
      <optional> <ref name="contact.address"/> </optional>
      <optional> <ref name="contact.information"/> </optional>
    </interleave>
    <attribute name="label"/>
  </element>
</define>
```

A contact group is a collection of references to contacts:

```
<define name="contacts.contactgroup">
  <element name="contactgroup">
    <zeroOrMore>
      <ref name="contactgroup.member"/>
    </zeroOrMore>
    <attribute name="label"/>
  </element>
</define>

<define name="contactgroup.member">
  <element name="member">
    <text/>
  </element>
</define>
```

A contact file is just a reference to a to be included file:

```
<define name="contacts.contactfile">
  <element name="contactfile">
    <text/>
  </element>
</define>
```

The fields that make up a contact are normally just text. The address needs to be preformatted with `<p>` elements (more details can be found in the file `x-corres.rng`).

A simple database will look like this:

```
<contacts>

  <contact label='hagen.j'>
    <initials>J.</initials>
    <formalname>Hagen</formalname>
    <informalname>Hans</informalname>
    <title>Drs.</title>
    <prefix>Heer</prefix>
    <address>
      <p>J. Hagen</p>
      <p>PRAGMA ADE</p>
      <p>Ridderstraat 27</p>
      <p>8061GH Hasselt NL</p>
    </address>
  </contact>

  <contact label='otten.a.f'>
    <initials>J.</initials>
    <formalname>Otten</formalname>
    <informalname>Ton</informalname>
    <title>Drs.</title>
    <prefix>Heer</prefix>
    <address>
      <p>A.F. Otten</p>
      <p>PRAGMA ADE</p>
      <p>Ridderstraat 27</p>
      <p>8061GH Hasselt NL</p>
    </address>
  </contact>

  <contact label='marle.k.van'>
    <initials>K. van</initials>
    <formalname>Marle</formalname>
    <informalname>Kees</informalname>
    <title>Ir.</title>
    <prefix>Heer van</prefix>
    <address>
      <p>K. van Marle</p>
      <p>PRAGMA POD</p>
      <p>Ridderstraat 27</p>
      <p>8061GH Hasselt NL</p>
    </address>
  </contact>

  <contactgroup label='pragma-ade'>
    <member>hagen.j</member>
    <member>otten.a.f</member>
  </contactgroup>
```

```
  <contactgroup label='pragma-pod'>
    <member>hagen.j</member>
    <member>otten.a.f</member>
    <member>marle.k.van</member>
  </contactgroup>

</contacts>
```

The labels plays an important role later on, when we will filter data from the database, so they should be unique and consistent.

3 The Interface

You can use the database from within ConTeXt after you've loaded the file `x-corres.tex`:

```
\usemodule[corres]
```

The macro that deals with processing is:

```
\XMLprocesscontacts[filename]
```

The argument is optional, in which case the selection needs to be set up with the following command:

```
\setvariables
  [contacts]
  [selection=,
   file=]
```

This command is also used for specifying the data file.

The fields of a contact are stored and can be used in the typesetting process. This process is hooked into the XML engine and style by the following setup (here we show the default definition):

```
\startsetups[contact:handle]

  \XMLflush{address}

\stopsetups
```

And so we have arrived at the main process, which may look like something:

```
\setvariables [contacts] [selection=pragma-ade,file=pragma.xml]

\startsetups [contact:handle]
  \XMLflush{initials} \XMLflush{formalname} \endgraf
\stopsetups

\XMLprocesscontacts
```

A selection can be a comma separated list of labels and/or groups. In a first pass, the list is resolved into a list of valid contacts, while the second pass typesets the result based on the handle.

4 The Style

In the previous chapters we presented the XML database and its interface, here will will spend some words on typesetting a (series of) letter(s). The style is quite independent of the database but uses the same fields and is called `m-letter.tex`.

It does not make sense to discuss all the details of the style because you need to look into the file anyway, so we stick to discussing the methodology used. There is an EXAMPLE interface that permits you to play with the settings.

The layout setup is rather simple and uses standard ConTEXt setup commands. The individual components that make up the letter are defined in so called setups, like:

```
\startsetups [letter:place:address]
  ...
\stopsetups
```

We use the layering mechanism to collect and place the components. Inside such a setup, variables are used to control the behaviour. For instance, the location of the address on the page grid is determined by:

```
\setvariables
  [letter:address]
  [line=3]
```

The address itself again a setup:

```
\setsetups[letter:address]
  ...
\stopsetups
```

and although a definition like:

```
\startsetups[letter:address]
  A Fancy Name \endgraf
  A Nice Address \endgraf
  The Place To Go
\stopsetups
```

is okay, it makes more sense to use variables again, like in:

```
\startsetups[letter:address]
  \def\\{\endgraf}\getvariable{letter:data}{address}
\stopsetups
```

Of course you need to define the adress:

```
\setvariables
  [letter:data]
  [address={Name\\Address\\City}]
```

This chaining of settings may look overly complicated, but the advantage is that by using setups we can easily overload components.

For the same reason, we have isolated the labels for instance:

```
\setuplabeltext
  [en]
  [letter:opening:formal=Dear,
   letter:opening:informal=Dear,
   letter:opening:unknown=LS]
```

Labels are defined per language, so that the style automatically adapts itself.

Now, when you load the style, and optionally overload setups, and definitely set the data variables typesetting the letter only takes a few lines of code.

```
\starttext
\setups [letter:place]
\stoptext
```

Interfacing to the XML database takes the following steps:

```
\setvariables
  [letter:data]
  [   address=\XMLflush{address},
       prefix=\XMLflush{prefix},
       suffix=\XMLflush{suffix},
     initials=\XMLflush{initials},
   formalname=\XMLflush{formalname},
 informalname=\XMLflush{informalname}]
      content=\getbuffer]
```

Here we assume that the content goes into a buffer. Of course we need to set the selection and filename for the contacts handler:

```
\setvariables
  [contacts]
  [selection=...,
   file=...]
```

Next we hook the letter style into the contacts handler:

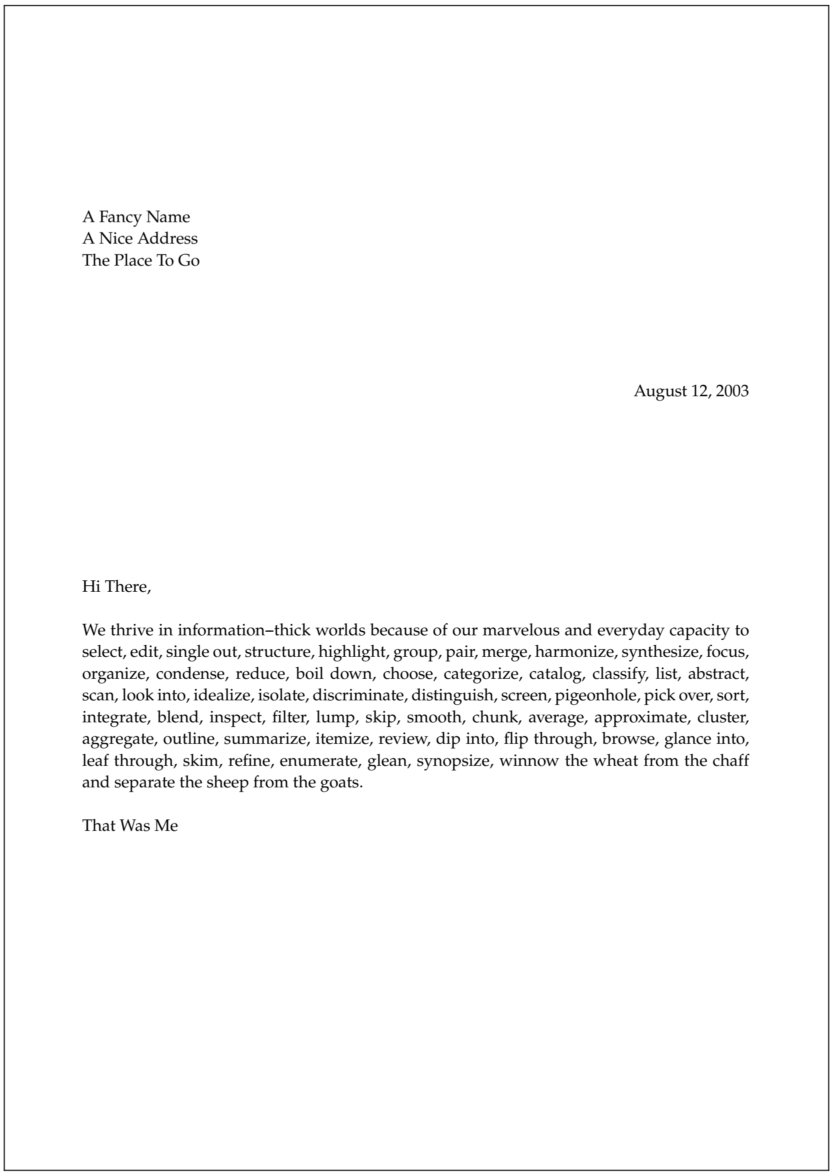
```
\startsetups[contact:handle]
  \setups[letter:place]
\stopsetups
```

And now we're ready to process the database and turn the addresses that match our selection criterium into a personalized letter.

```
\XMLprocesscontacts
```

An example of this integration can be found in the eXaMpLe module `x-e-0302.tex`.

5 Examples


A Fancy Name
A Nice Address
The Place To Go

August 12, 2003

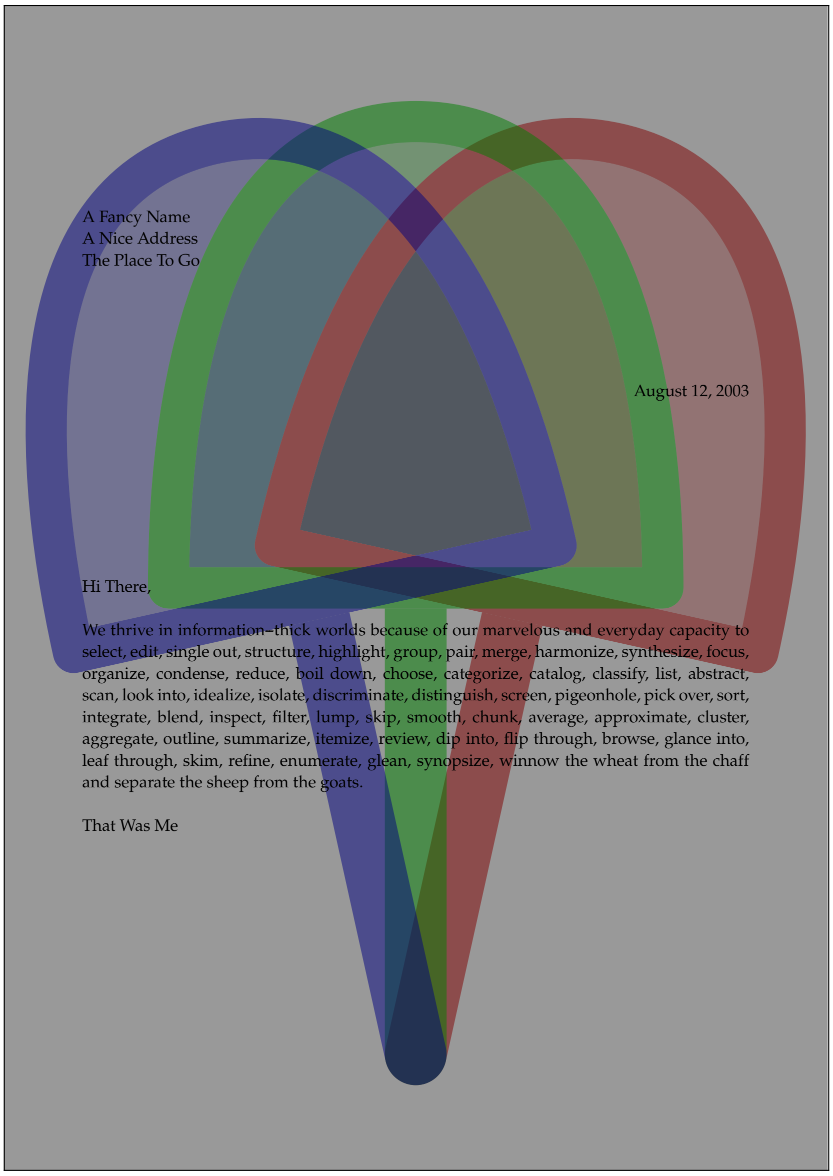
Hi There,

We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

That Was Me

```
\usemodule[letter]

\usetypescript
  [palatino]
  [\defaultencoding]

\setupbodyfont
  [palatino]

\setups[letter:test]

\starttext

\setups[letter:place]

\stoptext
```

In this example we choose a font and typeset a simple test letter. We use a test setup to initialize the different elements that make up a letter.


A Fancy Name
A Nice Address
The Place To Go

August 12, 2003

Hi There,

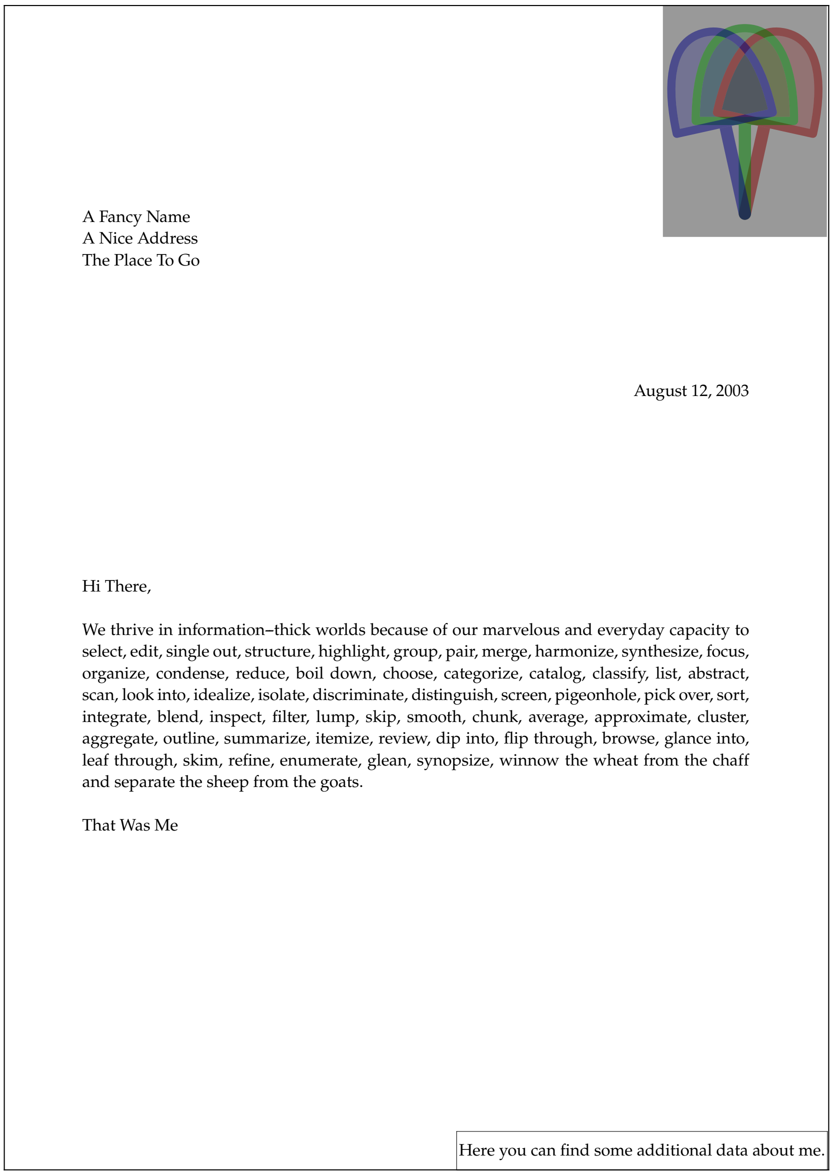
We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

That Was Me

```
\usemodule[letter]

\usetypescript
  [palatino]
  [\defaultencoding]

\setupbodyfont
  [palatino]

\setups[letter:test]

\setvariables
  [letter:head]
  [graphic=corrlogo.pdf]

\starttext

\setups[letter:place]

\stoptext
```

One of the first things you will like to do is to add a logo or other personal mark. There are several ways to do this. Here we put a graphic in the background.

This option makes sense if your logo is placed on a paper format similar to the one of your letter.



A Fancy Name
A Nice Address
The Place To Go

August 12, 2003

Hi There,

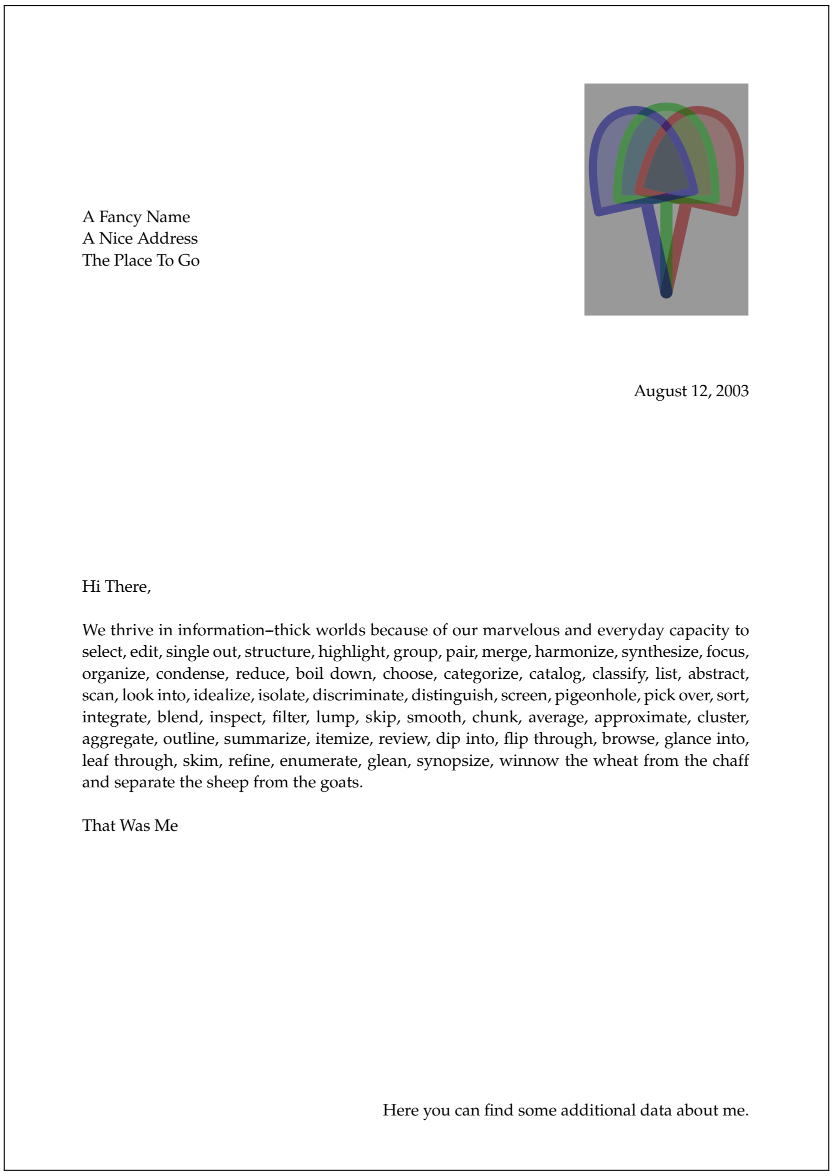
We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

That Was Me

Here you can find some additional data about me.

```
\usemodule[letter]

\usetypescript
  [palatino]
  [\defaultencoding]

\setupbodyfont
  [palatino]

\setups[letter:test]

\setlayer
  [letterhead]
  [preset=righttop]
  {\externalfigure
     [corrlogo.pdf]
     [width=.2\paperwidth]}

\setlayerframed
  [letterhead]
  [preset=rightbottom]
  [height=1cm]
  {Here you can find some
   additional data about me.}

\starttext

\setups[letter:place]

\stoptext
```

There is a special layer `letterhead` that you can use to position a graphic (or any content you want). You can place multiple graphic (or text) elements.



A Fancy Name
A Nice Address
The Place To Go

August 12, 2003

Hi There,

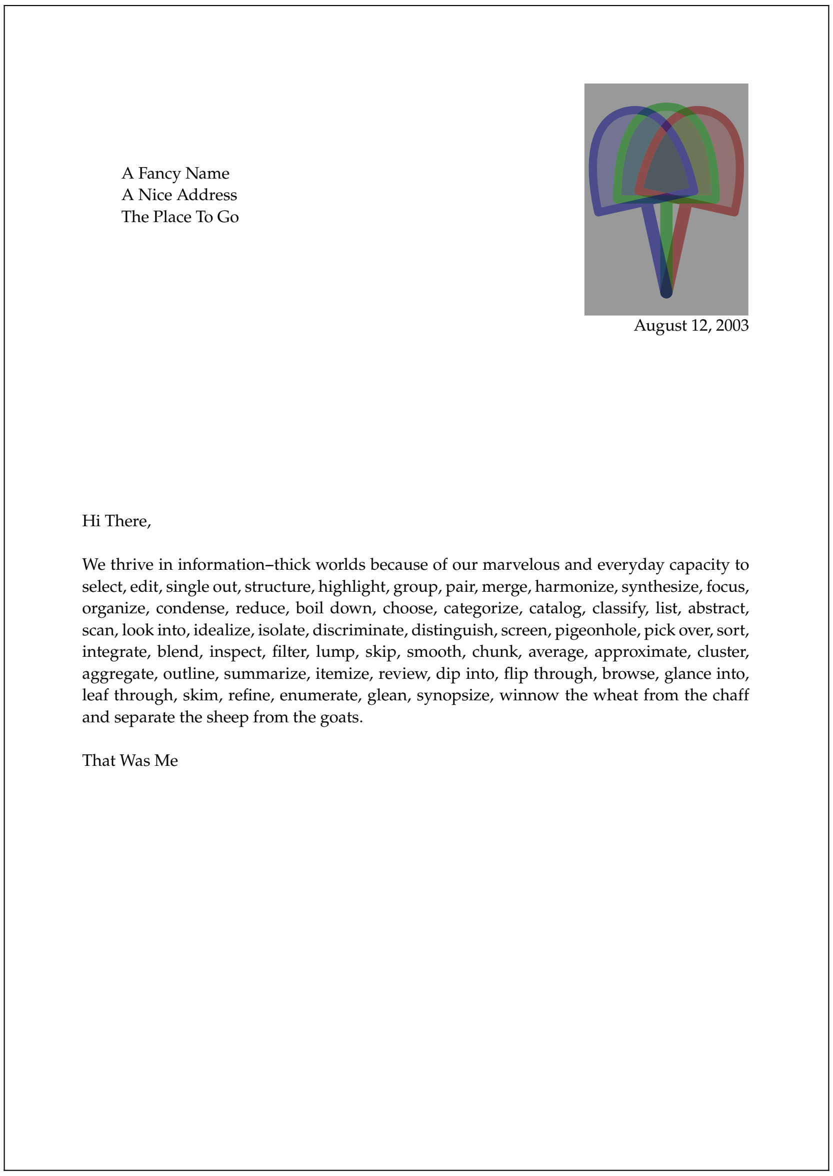
We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

That Was Me

Here you can find some additional data about me.

```
\usemodule[letter]

\usetypescript
  [palatino]
  [\defaultencoding]

\setupbodyfont
  [palatino]

\setups[letter:test]

\setlayer
  [letterhead]
  [preset=righttop,
   offset=\backspace]
  {\externalfigure
     [corrlogo.pdf]
     [width=.2\paperwidth]}

\setlayerframed
  [letterhead]
  [preset=rightbottom,
   hoffset=\backspace,
   voffset=.5\backspace]
  [height=1cm,
   offset=overlay,
   frame=off,
   strut=yes]
  {Here you can find some
   additional data about me.}

\starttext

\setups[letter:place]

\stoptext
```

Layers permits you to finetune the position, for instance with the `(v|h)offset` keys. The offsets are relative to the corner. More information about layers can be found in the 'details' manual.



A Fancy Name
A Nice Address
The Place To Go

August 12, 2003

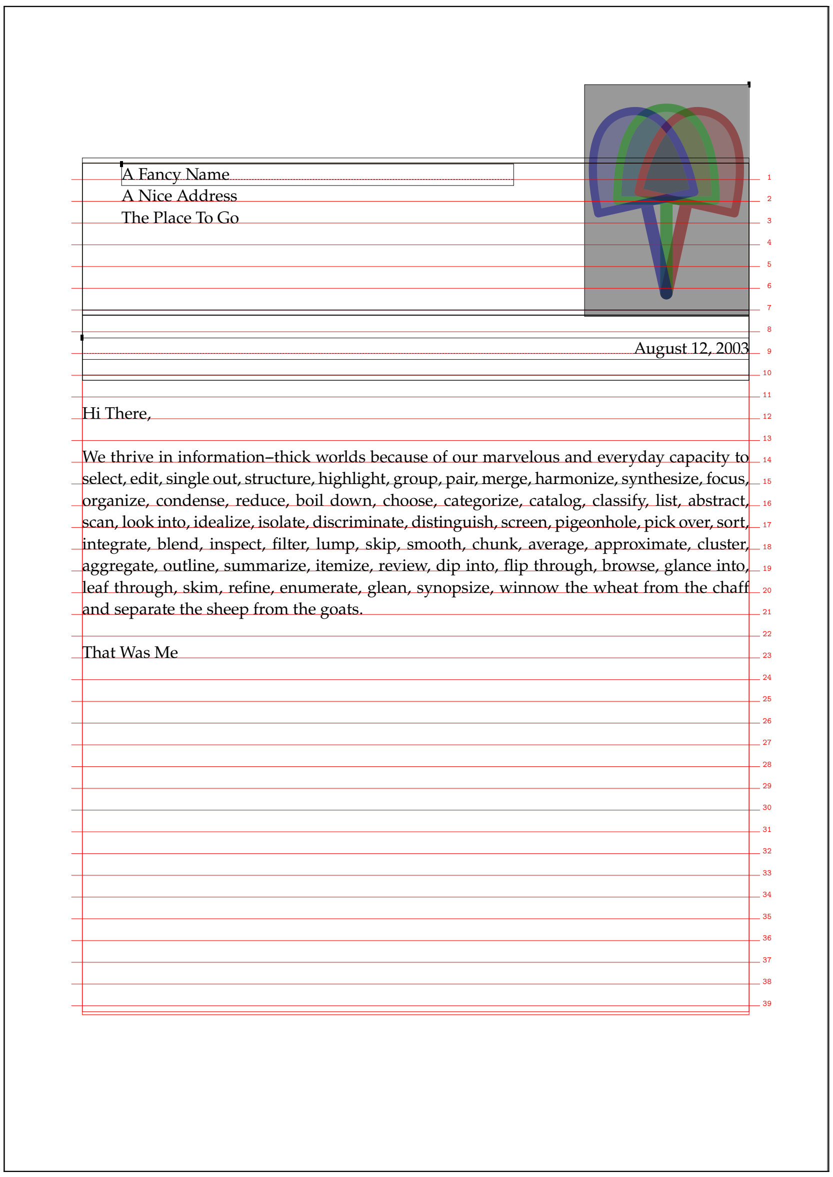
Hi There,

We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

That Was Me

```
\usemodule[letter]

\usetypescript
  [palatino]
  [\defaultencoding]

\setupbodyfont
  [palatino]

\setups[letter:test]

\setlayer
  [letterhead]
  [preset=righttop,
   offset=\backspace]
  {\externalfigure
     [corrlogo.pdf]
     [width=.2\paperwidth]}

\setvariables
  [letter:address]
  [line=1,
   hoffset=1cm,
   width=10cm,
   noflines=7]

\starttext

\setups[letter:place]

\stoptext
```

The adress and reference elements are also positioned in layers, but this time they are not in the background but flushed in the normal text stream. Their content tries to honor the grid. Instead of manipulating the layers directly, we use intermediate variables.



A Fancy Name
A Nice Address
The Place To Go

August 12, 2003

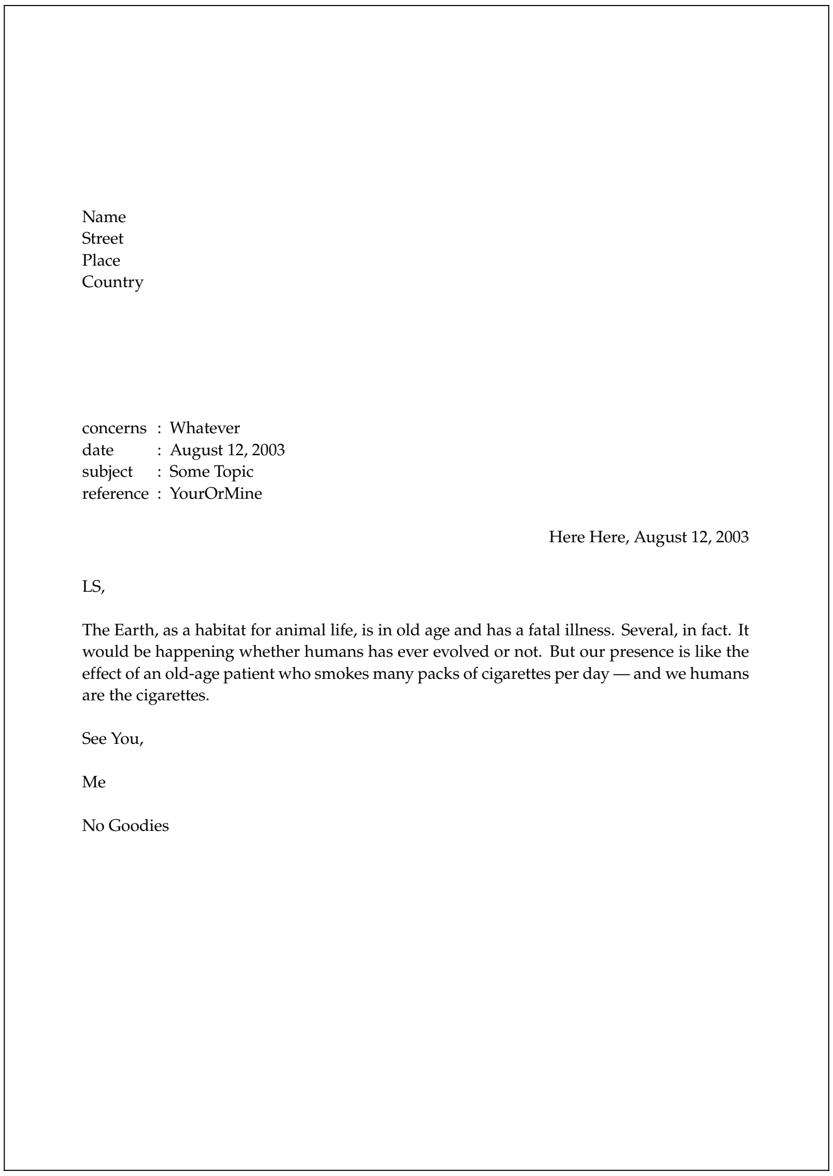
Hi There,

We thrive in information–thick worlds because of our marvelous and everyday capacity to select, edit, single out, structure, highlight, group, pair, merge, harmonize, synthesize, focus, organize, condense, reduce, boil down, choose, categorize, catalog, classify, list, abstract, scan, look into, idealize, isolate, discriminate, distinguish, screen, pigeonhole, pick over, sort, integrate, blend, inspect, filter, lump, skip, smooth, chunk, average, approximate, cluster, aggregate, outline, summarize, itemize, review, dip into, flip through, browse, glance into, leaf through, skim, refine, enumerate, glean, synopsize, winnow the wheat from the chaff and separate the sheep from the goats.

That Was Me

```
\showgrid
\showgridboxes
\tracelayerstrue

\usemodule[letter]

\usetypescript
  [palatino]
  [\defaultencoding]

\setupbodyfont
  [palatino]

\setups[letter:test]

\setlayer
  [letterhead]
  [preset=righttop,
   offset=\backspace]
  {\externalfigure
      [corrlogo.pdf]
      [width=.2\paperwidth]}

\setvariables
  [letter:address]
  [line=1,
   hoffset=1cm,
   width=10cm,
   noflines=7]

\setvariables
  [letter:reference]
  [line=2,
   noflines=3]

\starttext

\setups[letter:place]

\stoptext
```

The previous page showed some placement options but without cues this is not that easy to do. Therefore we have turned on some tracing options here.


Name
Street
Place
Country

concerns : Whatever
date : August 12, 2003
subject : Some Topic
reference : YourOrMine

Here Here, August 12, 2003

LS,

The Earth, as a habitat for animal life, is in old age and has a fatal illness. Several, in fact. It would be happening whether humans has ever evolved or not. But our presence is like the effect of an old-age patient who smokes many packs of cigarettes per day — and we humans are the cigarettes.

See You,

Me

No Goodies

```
\usemodule[letter]

\usetypescript
  [palatino]
  [\defaultencoding]

\setupbodyfont
  [palatino]

\setvariables
  [letter:data]
  [address={Name\\Street\\
    Place\\Country}]

\setvariables
  [letter:data]
  [subject=Some Topic,
   date=\currentdate,
   concerns=Whatever,
   residence=Here Here,
   reference=YourOrMine]

\startbuffer[texletter]
\input ward
\stopbuffer

\setvariables
  [letter:data]
  [author=Me,
   greeting=See You]

\setvariables
  [letter:data]
  [appendices=No Goodies]

\starttext

\setups[letter:place]

\stoptext
```

In addition to some style manipulation variables, we also use variables for setting the actual content of the letter.

todo: style key

todo: no modes, vars

6 The GUI

to do

7 The Tools

We are in the process of writing an associated tool for sorting, cleaning up, and manipulating the correspondence database. This tool will be part of the ConTeXt distribution as soon as it's stable and documented.