This Way

ConTEXt magazine #6
June 2004

A Few Dangerous Features

Hans Hagen
PRAGMA ADE

Occasionally we experiment a bit with (PDF) features that are useful but at the same time dangerous when applied uncontrolled. In the process of cleaning up some files in my source tree and triggered by a discussion about overprint I decided to move some of that code into the kernel. You are warned!

Remark

The features discussed here have a so called global character, i.e. all settings are global by nature. Future releases may introduce (and by default change to) local behaviour. So, don't make your documents depending on local/global behaviour. In most cases you will probably not notice the difference.

Being negative

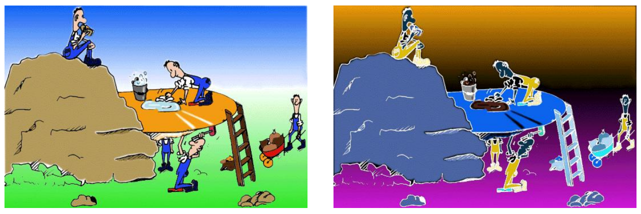
The ConTEXt page imposition machinery provides negation because sometimes raster image processors need that feature. In that case negation is applied to the whole page. Within the document stream inverted colors are normally (and best) realized with defining an appropriate color. For special purposes we also provide negation

```
\startcolor[red]\ignorespaces
    \input ward
    \startnegative\ignorespaces
        \input ward
        \startpositive\ignorespaces
            \input ward
        \removeunwantedspaces\stoppositive
        \input ward
    \removeunwantedspaces\stopnegative
    \input ward
\removeunwantedspaces\stopcolor
```

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 had 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. 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 had 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. 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 had 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.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 had 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.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 had 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.

We can also apply negation to graphic, but the result may not be what we expect. While writing this document figure 1 negates well when view in GHOSTSCRIPT but ACROBAT 6 shows a strange vertical line pattern.

```
\startcombination
  {\startpositive
     \externalfigure[hacker.jpg][width=4cm]%
   \stoppositive}
  {normal}
  {\startnegative
     \externalfigure[hacker.jpg][width=4cm]%
   \stopnegative}
  {negative}
\stopcombination
```



normal negative

Figure 1 Negation of graphics.

Font effects

Another bag of tricks concerns font effects. As with negation and the to be discussed overprint these are implemented using the CONTEXT (still experimental) feature handler, but this time we don't provide direct commands. Instead we use arguments to control the effects.

```
In this paragraph we have \starteffect[hidden]hidden a piece of
text\stopeffect. How useful this feature is depends on the kind of
documents you make. An alternative is to put the text in a viewer
layer (\starteffect[hidden]as provided by \PDF\stopeffect) that is
hidden, but since that feature is not widely available the effects
approach is safer.
```

In this paragraph we have   . How useful this feature is depends on the kind of documents you make. An alternative is to put the text in a viewer layer (   ) that is hidden, but since that feature is not widely available the effects approach is safer.

More interesting is changing the way a font is rendered. An outline version is rendered with the outer effect.

```
\bf \starteffect[outer]\processfile{ward}\stopeffect \par
```

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 had 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.

The inner effect is the normal one so there is no reason to show it here. The both option combines the two resulting in an extra bold version.

```
\bf \starteffect[both]\processfile{ward}\stopeffect \par
```

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 had 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.

You can influence the linewidth as is demonstrated in the following example:

```
\setupproperty[outer][rulethickness=.8pt]
\bfd \starteffect[outer]Bigger is Beautiful\stopeffect
```

Bigger is Beautiful

Speaking of 2004, in ConTeXt (read: TeX) intercharacter spacing can only be achieved by macro processing. The next method works well, but you need to manipulate the \hsize yourself, since the typesetting engine is unaware of this backend manipulation.

```
\setupproperty[both][stretch=2]
\setupalign[right]
\dontleavehmode \hsize=.6\hsize
\bf \starteffect[both]\processfile{ward}\stopeffect \par
```

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 had 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.

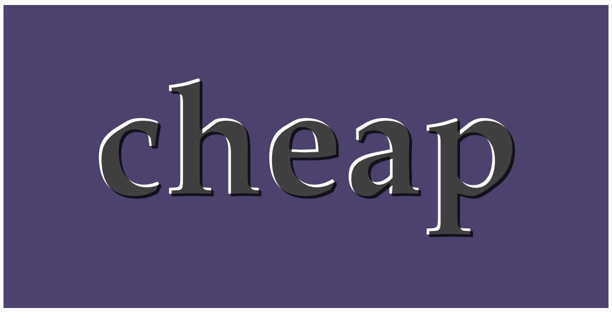
The normal (or inner) alternative looks as follows:

```
\setupproperty[normal][stretch=2]
\setupalign[right]
\dontleavehmode \hsize=.6\hsize
\bf \starteffect[normal]\processfile{ward}\stopeffect \par
```

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 had 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.

Overprint and knockout

Another feature that should be used with care is overprint. Normally a raster image processor will knock out colored areas under colored text or areas on top. This works well when the printing engine (or press) is able to precisely align the color plates. If not, you will get artifacts that show up as follows (often such effects occur in newspapers and cheap magazines):



On the one hand we get white spots and depending on how well the ink covers, we can get darker spots as well. In such cases it's best to overprint the background, which of course only works as expected when the top color is a well covering black. Otherwise we probably may have to compensate the color, which in turn depends on the kind of paper used.

At the document level, you can set the overprint with:

```
\setupcolors[overprint=yes]
```

We show a few examples of local usage: a simple application first (figure 2a):

```
\framed
  [background=color,backgroundcolor=ocyan,
   frame=off,offset=.25cm,strut=no]
  {\bfb\setstrut
   \startoverprint
   \framed
     [background=color,backgroundcolor=omagenta,
      foregroundcolor=oyellow,align={lohi,middle},
      frame=off,width=2.5cm,height=2cm]
     {overprint\\\property[knockout]{knockout}}%
   \stopoverprint
   \framed
     [background=color,backgroundcolor=omagenta,
      foregroundcolor=oyellow,align={lohi,middle},
      frame=off,width=2.5cm,height=2cm]
     {knockout\\\property[overprint]{overprint}}}%
```

We can nest overprint and turn it off as well (figure 2b):

```
\startoverprint
\framed
  [background=color,backgroundcolor=ocyan,
   frame=off,offset=.25cm,strut=no]
  {\bfb\setstrut
   \framed
     [background=color,backgroundcolor=omagenta,
      foregroundcolor=oyellow,align={lohi,middle},
      frame=off,width=2.5cm,height=2cm]
     {overprint\\\property[knockout]{knockout}}%
   \startknockout
   \framed
     [background=color,backgroundcolor=omagenta,
      foregroundcolor=oyellow,align={lohi,middle},
      frame=off,width=2.5cm,height=2cm]
     {knockout\\\property[overprint]{overprint}}%
   \stopknockout}%
\stopoverprint
```

Sometimes the overprint preview in ACROBAT works better when we apply a gray background (figure 2c). We use rather ugly pure CMYK colors, otherwise the effect is not visible in overprint preview mode.

```
\framed
  [background=color,backgroundcolor=ogray,backgroundoffset=.25em,
   frame=off,offset=overlay]
  {\getbuffer[a]}
```

and (figure 2d):

```
\framed
  [background=color,backgroundcolor=ogray,backgroundoffset=.25em,
   frame=off,offset=overlay]
  {\getbuffer[b]}
```

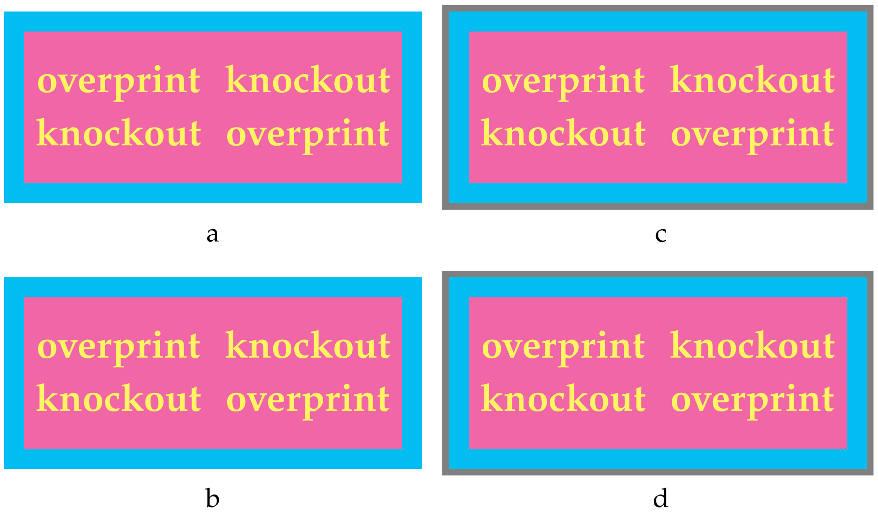


Figure 2 Preview overprint and knockout.

If we look at examples b and d of figure 2) in ACROBAT overprint preview mode, we will see that the effect depends on where we apply the overprint settings. As said, these are tricky features and should be used with care and understanding.

source code of this document

```
\usemodule[mag-01,abr-02]

\setupcolors[rgb=no,cmyk=yes]

\definecolor[red]      [c=.25,m=.75,y=.75,k=.25]
\definecolor[green]    [c=.75,m=.25,y=.75,k=.25]
\definecolor[blue]     [c=.75,m=.75,y=.25,k=.25]

\definecolor[tred]     [c=.25,m=.75,y=.75,k=.25,t=.5,a=1]
\definecolor[tgreen]   [c=.75,m=.25,y=.75,k=.25,t=.5,a=1]
\definecolor[tblue]    [c=.75,m=.75,y=.25,k=.25,t=.5,a=1]
\definecolor[tblack]   [s=0,t=.75,a=1]

\definecolor[ocyan]    [c=.75]
\definecolor[omagenta] [m=.75]
\definecolor[oyellow]  [y=.75]
\definecolor[ogray]    [s=.5]

\setvariables
  [magazine]
  [title={A Few Dangerous Features},
   author=Hans Hagen,
   affiliation=PRAGMA ADE,
   date=June 2004,
   number=6]

\startbuffer[abstract]
Occasionally we experiment a bit with (\PDF) features that are
useful but at the same time dangerous when applied uncontrolled. In
the process of cleaning up some files in my source tree and triggered
by a discussion about overprint I decided to move some of that code
into the kernel. You are warned!
\stopbuffer

\starttext \setups [titlepage] \setups [title]

\subject{Remark}

The features discussed here have a so called global character, i.e.
all settings are  global by nature. Future releases may introduce
(and by default change to) local behaviour. So, don't make your
documents depending on local/global behaviour. In most cases you
will probably not notice the difference.
```

```
\subject{Being negative}

The \CONTEXT\ page imposition machinery provides negation because
sometimes raster image processors need that feature. In that case
negation is applied to the whole page. Within the document stream
inverted colors are normally (and best) realized with defining an
appropriate color. For special purposes we also provide negation

\startbuffer
\startcolor[red]\ignorespaces
    \input ward
    \startnegative\ignorespaces
        \input ward
        \startpositive\ignorespaces
            \input ward
        \removeunwantedspaces\stoppositive
        \input ward
    \removeunwantedspaces\stopnegative
    \input ward
\removeunwantedspaces\stopcolor
\stopbuffer

\typebuffer \getbuffer

We can also apply negation to graphic, but the result may not
be what we expect. While writing this document \in {figure}
[fig:negated] negates well when view in \GHOSTSCRIPT\ but
\ACROBAT~6 shows a strange vertical line pattern.

\startbuffer
\startcombination
  {\startpositive
     \externalfigure[hacker.jpg][width=4cm]%
   \stoppositive}
  {normal}
  {\startnegative
     \externalfigure[hacker.jpg][width=4cm]%
   \stopnegative}
  {negative}
\stopcombination
\stopbuffer

\typebuffer
```

```
\placefigure
  [here] [fig:negated]
  {Negation of graphics.}
  {\getbuffer}

\subject{Font effects}

Another bag of tricks concerns font effects. As with negation and
the to be discussed overprint these are implemented using the
\CONTEXT\ (still experimental) feature handler, but this time we
don't provide direct commands. Instead we use arguments to control
the effects.

\startbuffer
In this paragraph we have \starteffect[hidden]hidden a piece of
text\stopeffect. How useful this feature is depends on the kind of
documents you make. An alternative is to put the text in a viewer
layer (\starteffect[hidden]as provided by \PDF\stopeffect) that is
hidden, but since that feature is not widely available the effects
approach is safer.
\stopbuffer

\typebuffer \getbuffer

More interesting is changing the way a font is rendered. An outline
version is rendered with the \type {outer} effect.

\startbuffer
\bf \starteffect[outer]\processfile{ward}\stopeffect \par
\stopbuffer

\typebuffer \start \getbuffer \stop

The \type {inner} effect is the normal one so there is no reason to
show it here. The \type {both} option combines the two resulting in
an extra bold version.

\startbuffer
\bf \starteffect[both]\processfile{ward}\stopeffect \par
\stopbuffer

\typebuffer \start \getbuffer \stop
```

```
You can influence the linewidth as is demonstrated in the
following example:

\startbuffer
\setupproperty[outer][rulethickness=.8pt]
\bfd \starteffect[outer]Bigger is Beautiful\stopeffect
\stopbuffer

\typebuffer \start \getbuffer \stop

Speaking of 2004, in \CONTEXT\ (read: \TEX) intercharacter  spacing
can only be achieved by macro processing. The next method  works
well, but you need to manipulate the \type {\hsize} yourself, since
the typesetting engine is unaware of this backend manipulation.

\startbuffer
\setupproperty[both][stretch=2]
\setupalign[right]
\dontleavehmode \hsize=.6\hsize
\bf \starteffect[both]\processfile{ward}\stopeffect \par
\stopbuffer

\typebuffer \start \getbuffer \stop

The \type {normal} (or \type {inner}) alternative looks as follows:

\startbuffer
\setupproperty[normal][stretch=2]
\setupalign[right]
\dontleavehmode \hsize=.6\hsize
\bf \starteffect[normal]\processfile{ward}\stopeffect \par
\stopbuffer

\typebuffer \start \getbuffer \stop

\subject{Overprint and knockout}

Another feature that should be used with care is overprint. Normally
a raster image processor will knock out colored areas under colored
text or areas on top. This works well when the printing engine (or
press) is able to precisely align the color plates. If not, you will
get artifacts that show up as follows (often such effects occur in
newspapers and cheap magazines):
```

source code of this document

```
\definelayer[fake][width=6cm,height=4cm]

\setlayerframed
  [fake]
  [preset=lefttop]
  [frame=off,width=8cm,height=4cm,
   background=color,backgroundcolor=blue,foregroundcolor=white]
  {\definedfont[SerifBold at 6\bodyfontsize]cheap}

\setlayerframed
  [fake]
  [preset=lefttop,offset=1pt]
  [frame=off,width=8cm,height=4cm,
   foregroundcolor=tblack]
  {\definedfont[SerifBold at 6\bodyfontsize]cheap}

\startbaselinecorrection
\tightlayer[fake]
\stopbaselinecorrection

On the one hand we get white spots and depending on how well the ink
covers, we can get darker spots as well. In such cases it's best to
overprint the background, which of course only works as expected when
the top color is a well covering black. Otherwise we probably may have
to compensate the color, which in turn depends on the kind of paper
used.

At the document level, you can set the overprint with:

\starttyping
\setupcolors[overprint=yes]
\stoptyping

We show a few examples of local usage: a simple application
first (\in {figure} {a} [fig:overprint]):

\startbuffer[a]
\framed
  [background=color,backgroundcolor=ocyan,
   frame=off,offset=.25cm,strut=no]
  {\bfb\setstrut
   \startoverprint
   \framed
     [background=color,backgroundcolor=omagenta,
      foregroundcolor=oyellow,align={lohi,middle},
      frame=off,width=2.5cm,height=2cm]
     {overprint\\\property[knockout]{knockout}}%
   \stopoverprint
   \framed
     [background=color,backgroundcolor=omagenta,
      foregroundcolor=oyellow,align={lohi,middle},
      frame=off,width=2.5cm,height=2cm]
     {knockout\\\property[overprint]{overprint}}}%
\stopbuffer

\typebuffer[a]

We can nest overprint and turn it off as well (\in {figure}
{b} [fig:overprint]):

\startbuffer[b]
\startoverprint
\framed
  [background=color,backgroundcolor=ocyan,
   frame=off,offset=.25cm,strut=no]
  {\bfb\setstrut
   \framed
     [background=color,backgroundcolor=omagenta,
      foregroundcolor=oyellow,align={lohi,middle},
      frame=off,width=2.5cm,height=2cm]
     {overprint\\\property[knockout]{knockout}}%
   \startknockout
   \framed
     [background=color,backgroundcolor=omagenta,
      foregroundcolor=oyellow,align={lohi,middle},
      frame=off,width=2.5cm,height=2cm]
     {knockout\\\property[overprint]{overprint}}%
   \stopknockout}%
\stopoverprint
\stopbuffer

\typebuffer[b]

Sometimes the overprint preview in \ACROBAT\ works better when we
apply a gray background (\in {figure} {c} [fig:overprint]). We use
rather ugly pure \CMYK\ colors, otherwise the effect is not
visible in overprint preview mode.
```

```
\startbuffer[c]
\framed
  [background=color,backgroundcolor=ogray,backgroundoffset=.25em,
   frame=off,offset=overlay]
  {\getbuffer[a]}
\stopbuffer

\typebuffer[c]

and (\in {figure} {d} [fig:overprint]):

\startbuffer[d]
\framed
  [background=color,backgroundcolor=ogray,backgroundoffset=.25em,
   frame=off,offset=overlay]
  {\getbuffer[b]}
\stopbuffer

\typebuffer[d]

\startbuffer
\startcombination[2*2]
  {\getbuffer[a]} {a}
  {\getbuffer[c]} {c}
  {\getbuffer[b]} {b}
  {\getbuffer[d]} {d}
\stopcombination
\stopbuffer

\placefigure
  [here] [fig:overprint]
  {Preview overprint and knockout.}
  {\getbuffer}

If we look at examples~b and~d of \in {figure} [fig:overprint]) in
\ACROBAT\ overprint preview mode, we will see that the effect
depends on where we apply the overprint settings. As said, these are
tricky features and should be used with care and understanding.

\setups [listing] \setups [lastpage] \stoptext
```

source code of this document