

This Way

ConTEXt magazine #1102 MKIV
July 2011

Annotated Verbatim
Hans Hagen
PRAGMA ADE

Annotating verbatim content is done using a mechanism called escaping. For such special cases it's often best to define a specific instance.

```
\definetyping
  [annotatedtyping]
  [escape=/,
   color=darkblue,
   before=,
   after=]

\startannotatedtyping
bla = test           /bgroup /sl oeps /egroup
                    /bgroup /bf some more /egroup
    | another test
    | somethingverylong /bgroup /it oeps /egroup
\stopannotatedtyping
```

```
bla = test           oeps
                    some more
    | another test
    | somethingverylong oeps
```

In this example the / now serves as an escape character. Of course you can also use the normal backslash but then you need to use a command to specify it.

```
\setuptyping
  [annotatedtyping]
  [escape=\letterbackslash]
```

Now we can say:

```
\startannotatedtyping
bla = test           \bgroup \sl oeps \egroup
                    \bgroup \bf some more \egroup
    | another test
    | somethingverylong \bgroup \it oeps \egroup
\stopannotatedtyping
```

and get:

```
bla = test           oeps
                    some more
    | another test
    | somethingverylong oeps
```

You can also define an end symbol:

```
\setuptyping
```

```
[annotatedtyping]
[escape={//,*},
color=darkblue]

\definestartstop
[cmt]
[style=\rm\bf]
```

Here the // starts the annotation and * ends it.

```
\startannotatedtyping
bla = test           // \black // \cmt{oeps} *
                   // \black // \cmt{some more} *
    | another test
    | somethingverylong // \black // \cmt{oeps} *
\stopannotatedtyping
```

Contrary to the first example, all text in the annotation is treated as T_EX input:

```
bla = test           // oeps
                   // some more
    | another test
    | somethingverylong // oeps
```

You can consider using more balanced tagging, as in:

```
\startannotatedtyping
bla = test           // \black // \cmt{oeps} *
                   // \black // \cmt{some more} *
    | another test
    | somethingverylong // \black // \cmt{oeps} *
\stopannotatedtyping
```

Watch how we limit the annotation to part of the text:

```
\startannotatedtyping
bla = test           << \rm\bf first >> test
                   << \rm\bf second >> test
    | test
    | somethingverylong << \rm\bf fourth >> test
\stopannotatedtyping
```

The test at the end of the lines is verbatim again.

```
bla = test           << \rm\bf first >> test
                   << \rm\bf second >> test
    | test
    | somethingverylong << \rm\bf fourth >> test
```

If no end symbol is given, the end of the line is used instead:

```
\setuptyping
  [annotatedtyping]
  [escape={//,},
   color=darkblue]
```

Watch out: here we use {//,} and not just // (which would trigger the escaped variant).

```
\startannotatedtyping
bla = test           // \black // \cmt{oeps}
                   // \black // \cmt{some more}
    | test
    | somethingverylong // \black // \cmt{oeps}
\stopannotatedtyping
```

The result is:

```
bla = test           // oeps
                   // some more
    | test
    | somethingverylong // oeps
```

This can also be done easier by abusing the style option of cmt:

```
\definestartstop
  [cmt]
  [color=black,
   style=\black //\rm\bf\space]
```

When we give:

```
\startannotatedtyping
bla = test           // \cmt{oeps}
                   // \cmt{some more}
    | test
    | somethingverylong // \cmt{oeps}
\stopannotatedtyping
```

We get:

```
bla = test           // oeps
                   // some more
    | test
    | somethingverylong // oeps
```

For cases like this, where we want to specify a somewhat detailed way to deal with a situation, we can use processors:¹

```
\defineprocessor
[escape]
[style=bold,
color=black,
left=(,right=)]
```

The previous definition of the annotation now becomes:

```
\setuptyping
[annotatedtyping]
[escape=escape->{//,},
color=darkblue]
```

This time no commands are needed in the annotation:

```
\startannotatedtyping
bla = test           // first
                    // second
    | test
    | somethingverylong // fourth
\stopannotatedtyping
```

The processor is applied to all text following the //. Spaces before the text are stripped.

```
bla = test           (first)
                    (second)
    | test
    | somethingverylong (fourth)
```

As some characters are special to T_EX, sometimes you need to escape the boundary sequence:

```
\defineprocessor
[myescape]
[style=\rm\bf,
color=black]

\setuptyping
[annotatedtyping]
[escape=myescape->{\letterhash\letterhash,},
color=darkgreen]
```

¹ More mechanisms in ConT_EXt MkIV will use that feature.

All text between the double hashes and the end of the line is now treated as annotation:

```
\startannotatedtyping
bla = test          ## first \bf test
                   ## second \sl test
                   | test
                   | somethingverylong ## third \it test
\stopannotatedtyping
```

So we get:

```
bla = test          first test
                   second test
                   | test
                   | somethingverylong third test
```

We can beautify \TeX commenting as follows:

```
\defineprocessor
[comment]
[style=\rm,
color=black,
left={\tttf\letterpercent\space}]

\setuptyping
[annotatedtyping]
[escape=comment->{\letterpercent\letterpercent,},
color=darkblue]
```

Here the double comments are turned into a single one and the text after it is typeset in a regular font:

```
\startannotatedtyping
bla = test          %% first \bf test
                   %% second \sl test
                   | test
                   | somethingverylong %% third \it test
\stopannotatedtyping
```

This gives:

```
bla = test          % first test
                   % second test
                   | test
                   | somethingverylong % third test
```

It is possible to define several escapes. Let's start with the delimited variant:

```

\defineprocessor
[escape_a]
[style=bold,
 color=darkred,
 left=(,
 right=)]

\defineprocessor
[escape_b]
[style=bold,
 color=darkgreen,
 left=(,
 right=)]

\setuptyping
[annotatedtyping]
[escape={escape_a->{[,]},escape_b->{[(,)]}},
 color=darkblue]

```

We can now alternate comments:

```

\startannotatedtyping
bla = test          [[ first ]] test [( first )]
                   [[ second ]] test [( second )]
    | test
    | somethingverylong [[ fourth ]] test [( fourth )]
\stopannotatedtyping

```

When typeset this looks as follows:

```

bla = test          (first) test (first)
                   (second) test (second)
    | test
    | somethingverylong (fourth) test (fourth)

```

The line terminated variant can also have multiple escapes.

```

\defineprocessor
[annotated_bf]
[style=\rm\bf,
 color=darkred]

\defineprocessor
[annotated_bs]
[style=\rm\bs,
 color=darkyellow]

```

```

\setuptyping
[annotatedtyping]
[escape={annotated_bf->{\bf,},annotated_bs->{\!bs,}},
color=darkblue]

```

So this time we have two ways to enter regular T_EX mode:

```

\startannotatedtyping
bla = test           !bf one {\em again}
                    !bs two {\em again}
    | test
    | somethingverylong !bf three {\em again}
\stopannotatedtyping

```

These somewhat meaningful tags result in:

```

bla = test           one again
                    two again
    | test
    | somethingverylong three again

```


source code of this document

```

% language=uk

% author      : Hans Hagen
% copyright   : PRAGMA ADE & ConTeXt Development Team
% license     : Creative Commons Attribution ShareAlike 4.0 International
% reference   : pragma-ade.nl | contextgarden.net | texlive (related) distributions
% origin      : the ConTeXt distribution
%
% comment     : Because this manual is distributed with TeX distributions it comes with a rather
%              liberal license. We try to adapt these documents to upgrades in the (sub)systems
%              that they describe. Using parts of the content otherwise can therefore conflict
%              with existing functionality and we cannot be held responsible for that. Many of
%              the manuals contain characteristic graphics and personal notes or examples that
%              make no sense when used out-of-context.

\usemodule[mag-01,abr-02]

\startbuffer[abstract]
  A not so widely known feature of the verbatim handler in \CONTEXT is the
  ability to add comments in another style and \MKIV even offers a bit more.
  Here some examples are shown.
\stopbuffer

\startdocument
  [title={Annotated Verbatim},
  author=Hans Hagen,
  affiliation=PRAGMA ADE,
  date=July 2011,
  number=1102 \MKIV]

\definetextbackground
  [example]
  [frame=on,
  framecolor=darkblue,
  location=paragraph,
  leftoffset=1ex,
  topoffset=1ex,
  bottomoffset=1ex]

Annotating verbatim content is done using a mechanism called escaping. For such
special cases it's often best to define a specific instance.

\startbuffer[define]
\definetyping
  [annotatedtyping]
  [escape=/,
  color=darkblue,
  before=,
  after=]
\stopbuffer

\startbuffer[example]
\startannotatedtyping
bla = test                /bgroup /sl oeps /egroup
                          /bgroup /bf some more /egroup

  | another test
  | somethingverylong /bgroup /it oeps /egroup
\stopannotatedtyping
\stopbuffer

```

source code of this document

```
\typebuffer[define,example][option=TEX] \getbuffer[define]

\starttextbackground[example]
  \getbuffer[example]
\stoptextbackground
```

In this example the `\type {/}` now serves as an escape character. Of course you can also use the normal backslash but then you need to use a command to specify it.

```
\startbuffer[setup]
\setuptyping
  [annotatedtyping]
  [escape=\letterbackslash]
\stopbuffer

\typebuffer[setup][option=TEX] \getbuffer[setup]
```

Now we can say:

```
\startbuffer[example]
\startannotatedtyping
bla = test          \bgroup \sl oeps \egroup
                   \bgroup \bf some more \egroup
  | another test
  | somethingverylong \bgroup \it oeps \egroup
\stopannotatedtyping
\stopbuffer

\typebuffer[example][option=TEX]
```

and get:

```
\starttextbackground[example]
  \getbuffer[example]
\stoptextbackground
```

You can also define an end symbol:

```
\startbuffer[setup]
\setuptyping
  [annotatedtyping]
  [escape={//,*},
  [color=darkblue]

\definestartstop
  [cmt]
  [style=\rm\bf]
\stopbuffer

\typebuffer[setup][option=TEX] \getbuffer[setup]
```

Here the `\type {//}` starts the annotation and `\type {*}` ends it.

```
\startbuffer[example]
\startannotatedtyping
bla = test          // \black // \cmt{oeps} *
                   // \black // \cmt{some more} *
  | another test
  | somethingverylong // \black // \cmt{oeps} *
\stopannotatedtyping
\stopbuffer
```

source code of this document

```
\typebuffer[example][option=TEX]
```

Contrary to the first example, all text in the annotation is treated as `\TEX\` input:

```
\starttextbackground[example]
  \getbuffer[example]
\stoptextbackground
```

You can consider using more balanced tagging, as in:

```
\startbuffer[setup]
\setuptyping
 [annotatedtyping]
 [escape={<<, >>}],
 [color=darkblue]
\stopbuffer
```

```
\typebuffer[example][option=TEX]
```

Watch how we limit the annotation to part of the text:

```
\startbuffer[example]
\startannotatedtyping
bla = test          << \rm\bf first >> test
                   << \rm\bf second >> test
    | test
    | somethingverylong << \rm\bf fourth >> test
\stopannotatedtyping
\stopbuffer
```

```
\typebuffer[example][option=TEX]
```

The `\type {test}` at the end of the lines is verbatim again.

```
\starttextbackground[example]
  \getbuffer[example]
\stoptextbackground
```

If no end symbol is given, the end of the line is used instead:

```
\startbuffer[setup]
\setuptyping
 [annotatedtyping]
 [escape={//,}],
 [color=darkblue]
\stopbuffer
```

```
\typebuffer[setup][option=TEX] \getbuffer[setup]
```

Watch out: here we use `\type {{//,}}` and not just `\type {//}` (which would trigger the escaped variant).

```
\definestartstop[cmt][style=\rm\bf]
```

```
\startbuffer[example]
\startannotatedtyping
bla = test          // \black // \cmt{oeps}
                   // \black // \cmt{some more}
    | test
    | somethingverylong // \black // \cmt{oeps}
\stopannotatedtyping
```

source code of this document

```
\stopbuffer
\typebuffer[example][option=TEX]
```

The result is:

```
\starttextbackground[example]
  \getbuffer[example]
\stoptextbackground
```

This can also be done easier by abusing the `\type {style}` option of `\type {cmt}`:

```
\startbuffer[setup]
\definestartstop
[cmt]
[color=black,
 style=\black //\rm\bf\space]
\stopbuffer

\typebuffer[setup][option=TEX] \getbuffer[setup]
```

When we give:

```
\startbuffer[example]
\startannotatedtyping
bla = test           // \cmt{oeps}
                    // \cmt{some more}
    | test
    | somethingverylong // \cmt{oeps}
\stopannotatedtyping
\stopbuffer

\typebuffer[example][option=TEX]
```

We get:

```
\starttextbackground[example]
  \getbuffer[example]
\stoptextbackground
```

For cases like this, where we want to specify a somewhat detailed way to deal with a situation, we can use processors: `\footnote {More mechanisms in \CONTEXT\MKIV\ will use that feature.}`

```
\startbuffer[setup]
\defineprocessor
[escape]
[style=bold,
 color=black,
 left=(,right=)]
\stopbuffer

\typebuffer[setup][option=TEX] \getbuffer[setup]
```

The previous definition of the annotation now becomes:

```
\startbuffer[setup]
\setuptyping
[annotatedtyping]
[escape=escape->{//,},
 color=darkblue]
\stopbuffer
```

source code of this document

```
\typebuffer[setup][option=TEX] \getbuffer[setup]
```

This time no commands are needed in the annotation:

```
\startbuffer[example]
\startannotatedtyping
bla = test           // first
                   // second
    | test
    | somethingverylong // fourth
\stopannotatedtyping
\stopbuffer
```

```
\typebuffer[example][option=TEX]
```

The processor is applied to all text following the `\type {//}`. Spaces before the text are stripped.

```
\starttextbackground[example]
  \getbuffer[example]
\stoptextbackground
```

As some characters are special to `\TEX`, sometimes you need to escape the boundary sequence:

```
\startbuffer[setup]
\defineprocessor
  [myscape]
  [style=\rm\tf,
   color=black]

\setuptyping
  [annotatedtyping]
  [escape=myscape->{\letterhash\letterhash,},
   color=darkgreen]
\stopbuffer
```

```
\typebuffer[setup][option=TEX] \getbuffer[setup]
```

All text between the double hashes and the end of the line is now treated as annotation:

```
\startbuffer[example]
\startannotatedtyping
bla = test           ## first \bf test
                   ## second \sl test
    | test
    | somethingverylong ## third \it test
\stopannotatedtyping
\stopbuffer
```

```
\typebuffer[example][option=TEX]
```

So we get:

```
\starttextbackground[example]
  \getbuffer[example]
\stoptextbackground
```

We can beautify `\TEX` commenting as follows:

```
\startbuffer[setup]
```

source code of this document

```

\defineprocessor
[comment]
[style=\rm,
color=black,
left={\tttf\letterpercent\space}]

\setuptyping
[annotatedtyping]
[escape=comment->{\letterpercent\letterpercent},]
color=darkblue]
\stopbuffer

\typebuffer[setup][option=TEX] \getbuffer[setup]

```

Here the double comments are turned into a single one and the text after it is typeset in a regular font:

```

\startbuffer[example]
\startannotatedtyping
bla = test          %% first \bf test
                   %% second \sl test
    | test
    | somethingverylong %% third \it test
\stopannotatedtyping
\stopbuffer

\typebuffer[example][option=TEX]

```

This gives:

```

\starttextbackground[example]
  \getbuffer[example]
\stoptextbackground

```

It is possible to define several escapes. Let's start with the delimited variant:

```

\startbuffer[setup]
\defineprocessor
[escape_a]
[style=bold,
color=darkred,
left=(,
right=)]

\defineprocessor
[escape_b]
[style=bold,
color=darkgreen,
left=(,
right=)]

\setuptyping
[annotatedtyping]
[escape={escape_a->{[[,]]},escape_b->{[(,)]}},]
color=darkblue]
\stopbuffer

\typebuffer[setup][option=TEX] \getbuffer[setup]

```

We can now alternate comments:

source code of this document

```
\startbuffer[example]
\startannotatedtyping
bla = test          [[ first ]] test [( first )]
                   [[ second ]] test [( second )]
    | test
    | somethingverylong [[ fourth ]] test [( fourth )]
\stopannotatedtyping
\stopbuffer
```

```
\typebuffer[example][option=TEX]
```

When typeset this looks as follows:

```
\starttextbackground[example]
  \getbuffer[example]
\stoptextbackground
```

The line terminated variant can also have multiple escapes.

```
\startbuffer[setup]
\defineprocessor
[annotated_bf]
[style=\rm\bf,
 color=darkred]

\defineprocessor
[annotated_bs]
[style=\rm\bs,
 color=darkyellow]

\setuptyping
[annotatedtyping]
[escape={annotated_bf->{\bf,},annotated_bs->{\bs,}},
 color=darkblue]
\stopbuffer

\typebuffer[setup][option=TEX] \getbuffer[setup]
```

So this time we have two ways to enter regular `\TEX\` mode:

```
\startbuffer[example]
\startannotatedtyping
bla = test          !bf one {\em again}
                   !bs two {\em again}
    | test
    | somethingverylong !bf three {\em again}
\stopannotatedtyping
\stopbuffer

\typebuffer[example][option=TEX]
```

These somewhat meaningful tags result in:

```
\starttextbackground[example]
  \getbuffer[example]
\stoptextbackground

\stopdocument
```

the *Journal of Applied Behavior Analysis* (JABA) and the *Journal of Experimental and Applied Behavior Analysis* (JEA).

The *Journal of Applied Behavior Analysis* is a peer-reviewed journal published quarterly by the Society for Applied Behavior Analysis. The journal is devoted to the publication of empirical research on behavior analysis and its applications. The journal is required reading for all behavior analysts and is considered one of the most important journals in the field. The journal is published by the Society for Applied Behavior Analysis, which is a professional organization of behavior analysts. The journal is published in the United States and is available to subscribers in other countries.

The *Journal of Experimental and Applied Behavior Analysis* is a peer-reviewed journal published quarterly by the Society for Applied Behavior Analysis. The journal is devoted to the publication of empirical research on behavior analysis and its applications. The journal is required reading for all behavior analysts and is considered one of the most important journals in the field. The journal is published by the Society for Applied Behavior Analysis, which is a professional organization of behavior analysts. The journal is published in the United States and is available to subscribers in other countries.

The *Journal of Applied Behavior Analysis* and the *Journal of Experimental and Applied Behavior Analysis* are both peer-reviewed journals published quarterly by the Society for Applied Behavior Analysis. The journals are devoted to the publication of empirical research on behavior analysis and its applications. The journals are required reading for all behavior analysts and are considered one of the most important journals in the field. The journals are published by the Society for Applied Behavior Analysis, which is a professional organization of behavior analysts. The journals are published in the United States and are available to subscribers in other countries.

The *Journal of Applied Behavior Analysis* and the *Journal of Experimental and Applied Behavior Analysis* are both peer-reviewed journals published quarterly by the Society for Applied Behavior Analysis. The journals are devoted to the publication of empirical research on behavior analysis and its applications. The journals are required reading for all behavior analysts and are considered one of the most important journals in the field. The journals are published by the Society for Applied Behavior Analysis, which is a professional organization of behavior analysts. The journals are published in the United States and are available to subscribers in other countries.

The *Journal of Applied Behavior Analysis* and the *Journal of Experimental and Applied Behavior Analysis* are both peer-reviewed journals published quarterly by the Society for Applied Behavior Analysis. The journals are devoted to the publication of empirical research on behavior analysis and its applications. The journals are required reading for all behavior analysts and are considered one of the most important journals in the field. The journals are published by the Society for Applied Behavior Analysis, which is a professional organization of behavior analysts. The journals are published in the United States and are available to subscribers in other countries.