

AcroT $\text{\E}$ X.Net

The AeB Pro Package  
Exploring the options for `\xBld`

D. P. Story

The L $\text{\A}$ T $\text{\E}$ X source file for this [AeB Blog](#) is attached to this document. Click [xbld\\_options.tex](#) to save the source file.

## 1. Options for `\xBld`

In AeB Pro the command that creates a layer is `\xBld`.<sup>1</sup> The syntax for this command is

```
\xBld[true|false|print=(true|false)]  
  {<layer_name>}<content in layer>\eBld
```

The layer is given a name `<layer_name>`, the content of the layer is enclosed in the `\xBld/\eBld` pair. The optional parameters are passed from within the brackets, and are described below:

- `true`: Inserting `true` as an optional argument causes the layer to be initially visible.
- `false`: Inserting `false` as an optional argument causes the layer to be initially hidden. The default is `false`, so if no `true` or `false` argument is passed as an optional argument, the layer will be initially hidden.
- `print`: This key allows you to set the print attribute of the layer: If `print=true`, the layer *always prints*, no matter if it is visible or not. If `print=false`, the layer *never prints*, no matter if it is visible or not. If the `print` key is not specified, then the layer will print if visible, otherwise,

---

<sup>1</sup>The primary demonstration file for `\xBld` is `aebpro_ex4.tex`, distributed with **AeB Pro**.

it does not print. Normally, the `print` key is not specified, and the layer is printed if visible.

In this article, we illustrate these optional parameters, the `print` key (requires `aeb_pro.dtx`, version `v1.0d`, `2009/08/07`, or later).

## 1.1. Examples

In this section we illustrate the optional parameters.

### • Visibility

Visibility of the layer is controlled by passing either `true` or `false` in the optional parameter list.

**Initially Visible.** This sentence is in its own layer, and its initial state is visible. **Toggle the text.** The source code for this paragraph follows:

```
\textbf{Initially Visible.}
\xBld[true]{initvisible}This sentence is in its own
layer, and its initial state is visible.\eBld
\setLinkText[\A{\JS{toggleSetThisLayer("initvisible");
}}]{\textcolor{red}{Toggle the text}}.
```

## Initially Hidden.

**Toggle the text.** The source code for this paragraph follows:

```
\xBld[false]{inithidden}This sentence is in its own  
layer, and its initial state is hidden.\eBld  
\setLinkText[\A{\JS{toggleSetThisLayer("inithidden");  
}}]{\textcolor{red}{Toggle the text}}.
```

### ● Set Print Attributes

**Always Print.** It may be that you want the layer to always print, even through it is initially hidden. Perhaps a logo hidden in some corner of the document, but prints when the document is printed. I won't bore you with some logo here; here's an example of a layer that always prints: I'm thinking about my formerly favorite number, can you guess its value?

Want a hint? **Click here.** I hope that hint worked for you. Click on the link to hide the layer again. The **answer** is .

The listing of this example follows, noting the optional parameter is `print`, which is the same as `print=true`. No visibility parameter is specified, so the layer is initially hidden, by default.

```
\xBld[print]{myformerfavnum}  
\fbox{The number is the value of the integral  
  
$$\int_0^4 4x + \frac{14}{dx} dx$$
.}\eBld  
Want a hint? \setLinkText[%  
\A{\JS{toggleSetThisLayer("myformerfavnum");  
}]]{\textcolor{red}{Click here.}}
```

**Never Print.** Now we can create a layer that takes up no space, but is overlaid on top of the tex content. We use the `textpos` commands to place the layer.

To view the hidden layer, [click here](#). You can use a form field to initiate the action to toggle the field as well

From this example you can see that links and forms appear to be handled differently: The layer, when visible, obscures the text of the link, whereas a form field is laid on top of the content of the page. Actually the link is laid on top of the content as well, it is only the text of the link this is obscured.

The code for the `textblock` environment can be seen from the source file, but we reproduce it here as well:

```
\begin{textblock*}{.67\linewidth}[.5,.5](.5\paperwidth,.8in)
\xBld[print=false]{acrotexrocks}%
\noindent\colorbox{red}{cornsilk}{%
\makebox[\linewidth-2\fbboxsep-2\fbboxrule][c]{\bfseries\LARGE
\color{blue}{Acro\negthinspace\TeX} rocks the world!}}\eBld
\end{textblock*}
```

You can see that the `print=false` parameter has been passed to the `\xBld`, to the mindless nonsense that is in the layer is never printed.

## 1.2. Rollovers

AeB Pro provides the command `\texHelp` to create a “rollover.” When the user rolls over a defined area, content that has been prepared in a hidden layer appears. As the user exits the defined area—the rollover area—the layer becomes hidden. The rollover area has a user defined rollover icon, defined by the command `\texHelpIndicator`, that signals to the user on the presence of a rollover.

The problem we address herein, is to define a rollover so that the content that appears never prints, and the rollover icon is in its own separate layer and it is set to visible but never prints.

Let’s now create a rollover,<sup>②</sup> shall we? The rollover icon is ‘<sup>②</sup>’, I’ve placed it in its own layer, made it visible, and set it to `print=false`.

The verbatim listings are

```
\begin{textblock*}{.67\linewidth}[.5, .5](.5\paperwidth, .25\paperheight)
\xBld[print=false]{rollover}\fcolorbox{red}{cornsilk}{%
\parbox{\linewidth-2\fbboxsep-2\fbboxrule}{Hello there,...}}\eBld
\end{textblock*}
```

Note the `\xBld[print=false]` above, this sets the content of the layer to never prints.

Below is a redefinition of the command `\texHelpIndicator` which defines the rollover icon.

```
\renewcommand{\texHelpIndicator}[1]{#1\thinspace$}^{\%  
  \xBld[true,print=false]{helpsym}\mbox  
  {\footnotesize\negthinspace\ding{109}}\raisebox{.225ex}  
  {\llap{\color{webblue}\tiny?\hspace{1.3ex}}}\eBld}$}
```

Again, we say `\xBld[true,print=false]`, the layer is initially on, and never prints.

Hope this file gives you idea on how you create your own layers with the attributes you want.