AcroTEX.Net

The AeB Pro Package
Layers and Rollovers

D. P. Story

http://www.acrotex.net
Published: January 12, 2007

1. Layers

When the `uselayers` option is taken, the necessary code is input to produce layers (Optional Content Groups). The AcroTEX Presentation Bundle (APB), which has a very sophisticated method of control over layers, by comparison, the AeB Pro layer support is very primitive indeed. As a rule, after you create a layer, you will need a control of that layer. This could be a button or a link that executes JavaScript.

The basic command for creating a layer is `\xBld`. The content of the layer is set off by the `\xBld`/`\eBld` pair. The command `\xBld` takes two parameters: (1) the first is optional, `true` if the layer is initially visible or `false`, the default, if the layer is hidden initially; (2) the name of the layer, this is used to reference the layer, to make it visible or hidden.

The creation and control of layers are illustrated in the two subsequent sections.

When constructing a layer, there are two possible scenarios:

1. The layer takes up “tex” space
2. The layer does not take up “tex” space

Let’s look at each of these in turn.

1.1. Layer takes up space

A layer that takes up space in tex is the easiest case. For example, guess what I’m thinking thinking about: I’m thinking about my formerly favorite number,

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

For those viewing this document in PDF, the relevant code code is seen below.

First the layer is enclosed in a `\xBld`/`\eBld` pair, the default state is hidden, so we don't supply the optional parameter. The require (second) parameter is the name used to refer to the layer.

The link text has a JavaScript action. First we get the OCG object for this layer by calling the `getxBld` function (this is part of the AeB Pro JavaScript) then if non-null (you may not have spelled the name correctly) we toggle the current state, `oLayer.state = !oLayer.state`.

```
\xBld{mythoughts}the number is the value of the integral
$\int_0^4 4x + \frac14\,dx$. \eBld Want a hint?
\setLinkText[%
\A{\JS{%
    var oLayer = getxBld("mythoughts");
    if ( oLayer != null )
        oLayer.state = !oLayer.state;
    }}
]{\textcolor{red}{Click here}}
```

An advantage of the layers approach is that the content of the layers are latexed as part of the content of the tex file; consequently, you can include virtually anything in your layer that tex can handle, math, figures, PSTricks, etc. Acrobat Pro 7.0 (and distiller) or later is required to build the layers, but only Adobe Reader 7.0 or later is needed to view the document, once constructed.

1.2. Layer takes no space

Perhaps the easiest way of creating layers that do not take up any tex space is to use a package such as textpos. In the preamble of this document, I've placed the following command:

```
\usepackage{pstricks-add,pstricks}
\usepackage[absolute,overlay]{textpos}
```

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. Once created and placed, we must have a way of showing it and hiding it. This time, we'll use a button: Push

As you can see from this example, you must be very careful in the placement of your layers vis-à-vis the button. Form fields are laid on top of content, and a layer is considered part of the content. This particular layer could be move to the upper left corner, for example, Let's see how a link works, shall we? Click here! The link is active even through the layer, you just have to find it! I could have set this link to that it has a visible bounding box, in which case the border of the bounding box shows through the layer, but hey, at least you can find the link!

One thing you need to keep in mind is the placement of the code, within the `textblock` environment. The overlaid material will appear on whatever page the `textblock` ends up on. Should you are more material before the `textblock`, the controls may drop off to the next page. When the user presses the control, nothing happens, actually, the layer is appearing on the previous page. Sometimes it makes sense to insert a `\newpage` to ensure the proper placement of the overlay relative to the placement of the controls.

2. Rollovers

The AeB Pro package offers you two rollovers, which ostensibly provides help to the document consumer.

2.1. `\texHelp` uses Layers

The rollover is a special case of the techniques discussed in section 1.2. For a rollover, you create a form field, enclosing a word, when the user rolls across the word, a help box opens. For example, recall the graph of the $\sin(x)$,ⓘ isn't it pretty? Try rolling over the word that is tagged with the 'ⓘ' symbol.

You will have to look at the source file to see what goes on here, suffice it to say that the command `\texHelp` was used to create the rollover, the symbol that appears is defined in the aeb_pro package by `\texHelpIndicator`, this command can be re-defined to have a different symbol.

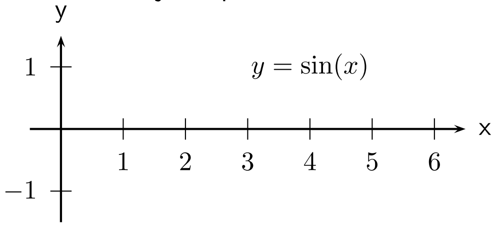
2.2. `\pdfHelp` does not

There is another type of rollover, `\pdfHelp`. The command takes three arguments, the first one is the name of the button field that is enclose around the third argument, the second argument is the help text. The help text is not part of the LaTeX content of the document is is not compiled by the TeX compiler, so it should be text. For example, can you remember the inventor of TeX?ⓘ

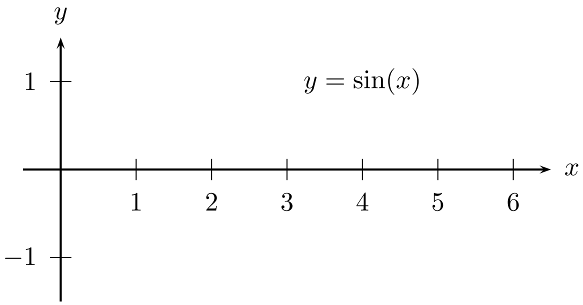
As with `\texHelp`, the symbol that tags the word as a help word is defined in aeb_pro.dtx by the command `\pdfHelpIndicator`. This command can be re-defined as needed.

3. Layers and Animation

Let's see if we can conjure up a little animation, shall we?



It is possible to combine the techniques of seciton 1.2 with those of animation, to create an anime that pops up. We leave this as an exercise for the reader, that's you.


y
x
1
-1
1
2
3
4
5
6
$y = \sin(x)$

Step Through