FlippyFloppy.eps ¬

Welcome to the FlippyFloppy ***BackSpace*** module.

You'll notice that this is set up as a **project**. This works remarkably well, it turns out, except for one minor annoyance. After ProjectBuilder has built the project, you'll have to rename to folder - simply cut out the *O.bundle* part and replace it with *Module* (so that the extension on the folder becomes *.BackModule*, silly!). It also means you can't just do a **make install**. You will have to make the project, rename the folder, and drag it to where you want it. If anybody knows how to make **ProjectBuilder** do the right thing here, let me know. Unfortunatley, the default make arguments are a preference in **ProjectBuilder** and are not part of the project itself. Otherwise, there would be an easy way.

This is a fun little module that does something simple. At least it's simple on a **Display**

PostScript system (plug, plug).

One of the more interesting aspects of the module is it's ability to let the user set the font for the text being flipped. As I thought about how to do this, it occurred to me there might be contention between different modules for the font panel. I figured out a way to solve this, and therefore propose the following **BackSpace Module FontPanel Protocol**.

- The first time a module wants to use the font panel, it must put itself on the ***BackSpace*** window's responder chain:

```
if (!inResponderChain) {
   [self setNextResponder:[[self window] nextResponder]];
   [[self window] setNextResponder:self];
inResponderChain = YES;
}
```

- The module must implement a method **chainFont<modulename>: sender**.   For example, **changeFontFlippyFloppy:**

- The module then, every time it wants the font panel, set the FontManager's action method to it's particular change font method:

```
[[FontManager new] setAction:@selector(changeFontFlippyFloppy:)];
```

In this way, every ***BackSpace*** module can share the font panel, without having to worry that if someone switches to another ***BackSpace*** module, uses the font panel there, and switches back, that the FontPanel will have the generic **changeFont:** method swallowed by the module that last put themselves on the responder chain for the ***BackSpace*** window. Even better, any module that implements this protocol is immune from those that don't, even though those that don't may screw each other up!

There is one quirk in this module that I really didn't feel like fixing (is was more trouble than it was worth, it seemed) - and that is in order to do mutiple lines of text, you must explicitly enter line endings with **Control-j**. Big deal, right?

It's cute, it's fun, it's frivolous, and X (eew!) can't even dream of doing it. Even with an R5 FontServer!

-Scott Byer

byer@adobe.com
December 3, 1992

Version 1.1, Dec 7th, 1992

- Fixed the setcachelimit call to restore the right cache limit upon leaving FlippyFloppy - the single ops library call for this operator is broken, thinking it nees a floating point value, when the operator only takes integer values, thus an attempt to use this singleop usually ends up shutting the font cache off or causing an error.
- Fixed wierd interactions between the font panel and other modules (most notably, Space). This was due to responder lists getting hosed and circles and junk being introduced and some sort of interaction with the above setcachelimit problem.