

After Dark™ About Box
A Library Function
for Implementing a
Scrolling Text About Box
in
After Dark Modules

What is it?

For After Dark module programmers using Symantec's THINK Pascal...

After Dark About Box (ADAB) is a library code resource for implementing a scrolling text window in an After Dark help box. You supply the PICT resource and the TEXT resource and ADAB provides your module with an about box featuring your text instructions scrolling on top of your picture. ADAB works in both black & white and color. It will display the text in any color you choose and will scroll text over your picture or over a blank or colored background of your choice. If there isn't enough memory to scroll text, ADAB will display a "standard" After Dark help box using alternate text you provide for just such an event.

How do I use it?

To use this library function you should have Berkeley Systems' After Dark Programming Examples and the After Dark 2.0 [ResEdit] TMPLs. You should, also, already be familiar with programming After Dark modules.

Procedure for using ADAB

1) Resources. Add the following resources to your resource file:

a. A black & white (1 bit) 'PICT' resource 243 pixels high by 222 pixels wide with ID *not* equal to 1000. This is the picture used in black & white mode. Please note that you should have no 'PICT' resources with ID = 1000. (Optional)

b. A color (8 bit) 'PICT' resource 243 pixels high and 222 pixels wide with

ID *not* equal to 1000. This is the picture used in color mode. Again, note that you should have no 'PICT' resources with ID = 1000. (Optional)

c. A 'Cals' resource with "DoAbout" selected (e.g. value = 1) with ID = 1. Please note that you will want DoDrawFrame, DoClose, DoBlank and DoInitialize selected as well. The 'Cals' resource tells your module what messages After Dark should send it. By selecting "DoAbout" you are letting After Dark know that your module will be handling the help box and AD should let your module know when the user clicks in the credits area.

d. A 'TEXT' resource with ID *not* equal to 1000. This text is the message ADAB will scroll over your picture. Please note that you should have no 'TEXT' resources with ID = 1000.

e. A second 'TEXT' resource with ID numerically one greater than that specified in "d," above and *not* equal to 1000. This is the text that will be displayed if there isn't enough memory to scroll the text in "d." You should make certain that it will fit in the help box. One way to do this is to temporarily change the ID of your scrolling 'TEXT' resource so the library function can't find it. It will then display the standard help box with this text.

2) Files. Add the following files to your project:

a. AD_About_Box.lib

b. AD_About_Box.p

3) Add code:

a. In your GraphicsModule_main.p add the following line to the "implementation" section:

```
function DoHelp (params: GMPParamBlockPtr): OSErr;  
external;
```

b. In your GraphicsModule_main.p add the following line to the function Main just after the call to "DoDrawFrame":

```
DoAbout:  
err := DoHelp(params);
```

c. Copy the function "DoHelp" from the sample "Generic Module.p" file supplied in the package to your own equivalent file in your module project. This is the file in which you have your "DoDrawFrame" function.

d. Be sure to declare the “DoHelp” function before the implementation section in your “Generic Module.p” file as follows:

```
function DoHelp (params: GMPParamBlockPtr): OSErr;
```

e. In the DoHelp function change the TEXTID, CPICTID and BWPICTID to match your TEXT resource, your color PICT resource and your black and white PICT resource, respectively. Also set your text color, background color, margins, etc.

Troubleshooting

If you've done everything above and when you click in the credits area and nothing happens or the standard text appears check the following:

- 1) Be sure no PICT or TEXT resources have ID = 1000.
- 2) Be sure that your PICT and TEXT IDs in your resource file agree with those you have passed in your DoHelp function.
- 3) The margins rects are for storing the margins, not the rect in which the text will appear. For instance, if CMargins.top equals 5 then the text will scroll up to 5 pixels from the top of the window before disappearing. Likewise if CMargins.bottom is zero then the text will appear to be coming up from the very bottom of the window. CMargins defines the margins for the color window and BWMargins for the black and white window.
- 4) If your text doesn't show up in color mode you may have set your text color to a value close to, but not equal to, pure white. You can either change the text color to equal pure white, which is quite a bit slower, or change it to a value further from white. Try a shade of gray. It will probably look like white against your picture anyway.

If you are wondering why this happens, it is because the system finds the color closest to the one you requested that it can display on the current device. If it happens to be white ADAB ends up writing white text on a white background in its buffer. When it comes time to copy the text to the window it finds nothing. If you request pure white for your text ADAB will use a black background in its buffer so it can distinguish text from background, but this will be slower.

Disclaimer

Use this software at your own risk. I make no warranties for it whatsoever.

Fee

ADAB is free to those who charge no fees for the software in which ADAB is incorporated. If you do charge a fee for your module, I expect you to pay me whatever you request for your module. Checks should be made out to:

David B. Zwiefelhofer
123 S. Randall Ave.
Madison, WI 53715-1360

After Dark™ About Box © 1993 subversive software.
123 s randall ave
madison, wi 53715-1360
internet: subversive@aol.com

Please send comments, bug reports, suggestions, et cetera.