Tutorial 3

What topics are covered?

- Asking the user a yes-no question.
- Displaying a message box.
- Optionally Aborting the installation.
- Displaying readme file upon install completion.
- Displaying advertising dialogs with 256 color bitmaps.

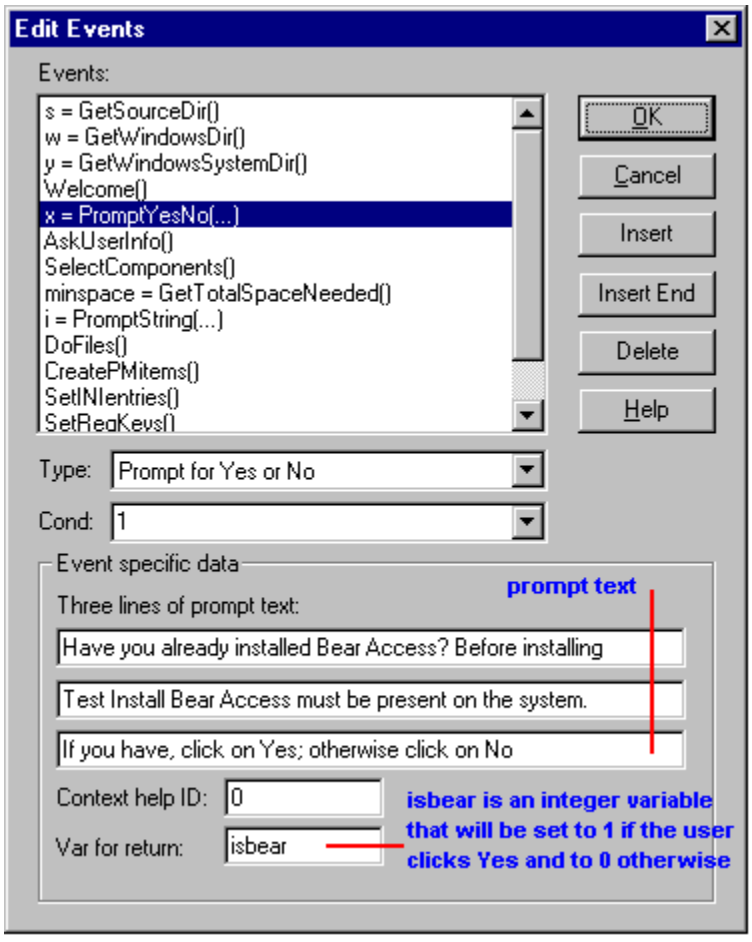
Step 1

Open c:\prj\install.inf. You will base this tutorial on the last one.

Step 2

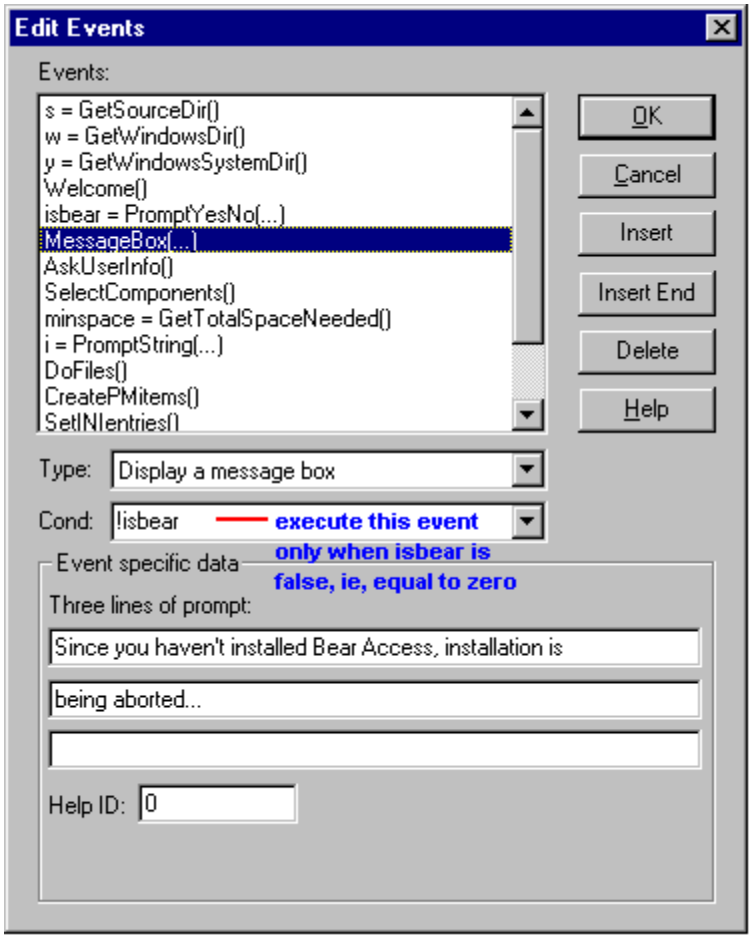
Suppose that you need to make sure the user has installed your another package before this one. If he/she hasn't, you'd like to display a message box and then abort the installation.

Go to the event list (Edit | Events), add a "prompt for yes/no" event and enter the parameters as shown below:



Step 3

If the user clicks on "No" then you will display a message box:



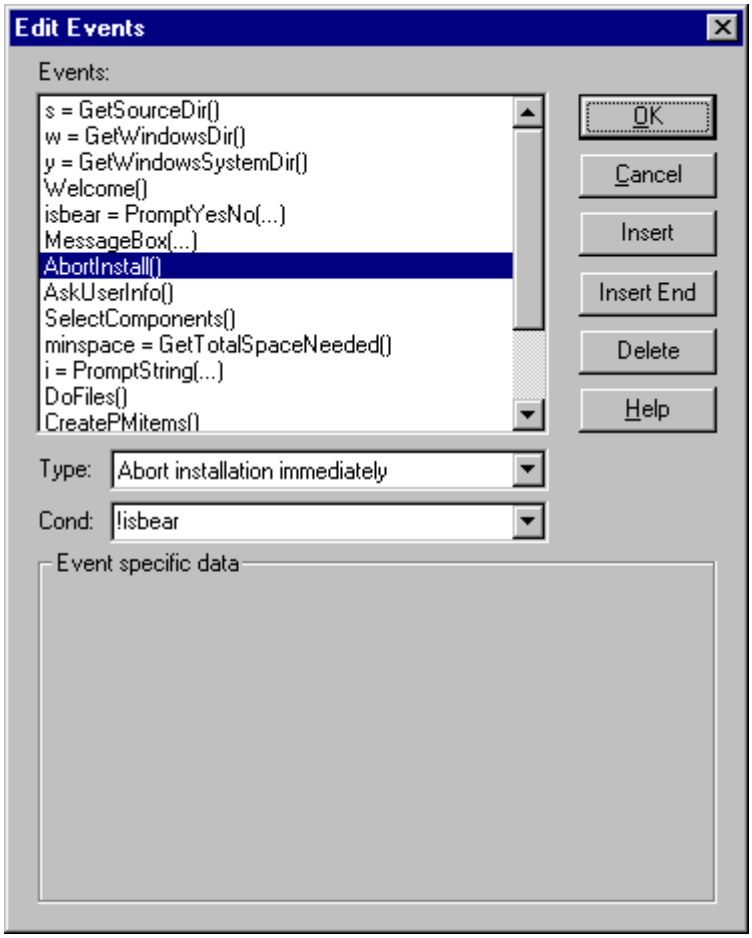
In the "cond" field there is an exclamation mark before "isbear". It means that the message box will be displayed only when isbear is false (i.e., 0). As an alternative we could have used "isbear == 0" (note the two consecutive =) instead of "!isbear".

If we wanted to display the message when isbear is true (i.e., non-0), we would simply put "isbear" in the "cond" field instead of "!isbear".

As a matter of fact, you can put complicated integer expression in the "cond" field, not just simple variables and logical negation. For example, you can put "x >= y && 2*z < 8" which reads "x is greater than or equal to y and 2 times z is less than 8".

Step 4

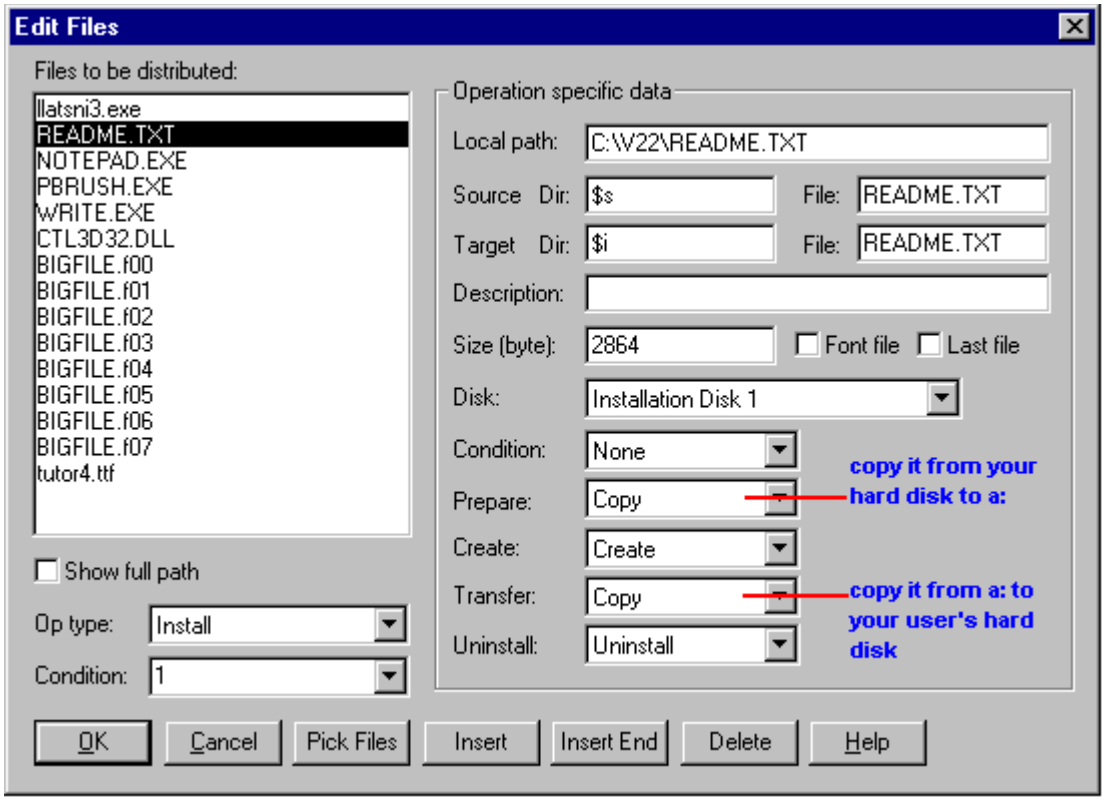
Add an "abort installation" event:



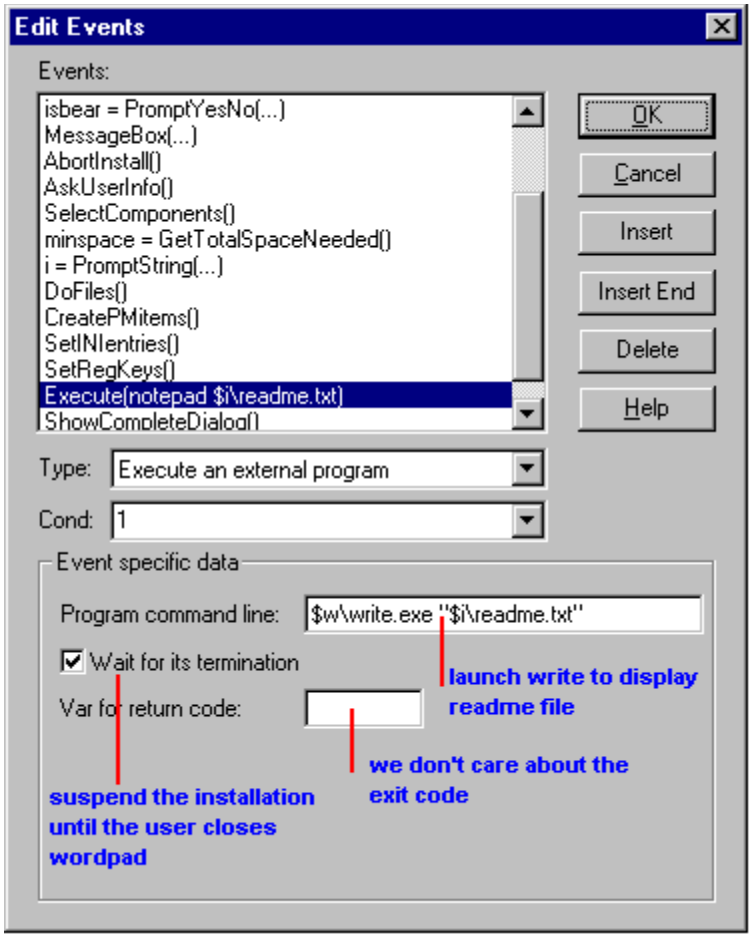
Note the "cond" field. It means that you will abort the installation only when isbear is false.

Step 5

Go to the file dialog and add an entry for readme.txt (find one on your own computer). Since you want to distribute it in uncompressed format, both "Prepare" and "Transfer" must be set to "Copy" at the same time. A common mistake is to set "Prepare" to "Copy" only and leave "Transfer" as "Decompress". Here is what the parameters should be like:



To display the readme file on installation completion, add an "execute extrernal program" event immediately before (or after, depending on your preference) the "show completion dialog" event as shown below:



Since it is difficult to locate wordpad (usually in c:\program files\accessories\), write, a wrapper of wordpad on Win95, is executed instead. It is easy to locate because it is always in the Windows directory, i.e., $w. However, this way whether "wait for its termination" is checked or not doesn't really matter since write terminates as soon as it launches wordpad.

Note that if your target platform is NT 3.5 or below then you shouldn't quote $i\readme.txt on the command line since the version of write.exe included in NT 3.5 doesn't seem to support quoted filenames. This means that this is generally a good idea not to use spaces in the default value of $i.

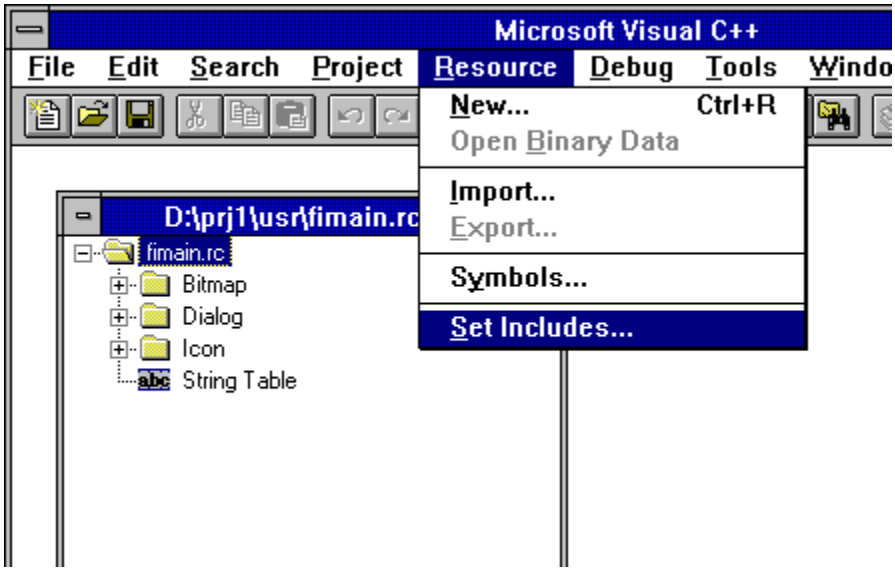
Click on "OK" button to close the dialog.

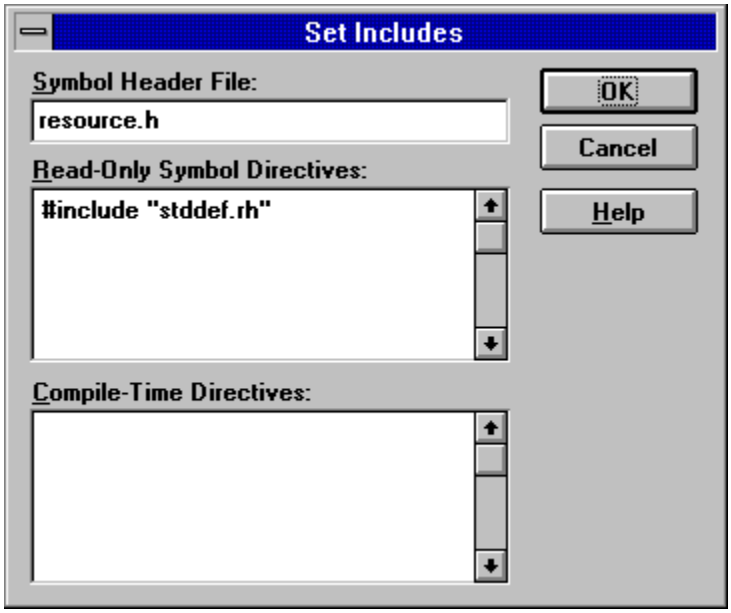
Step 6

Now, you are going to add an "advertising dialog" and put a 256 color bitmap in it. Please pay close attention. Many users have choked on this. To make use of this function, you must have the 32bit resource compiler (brcc32.exe and related files for BC4.5; rc.exe, link.exe and related files for VC2.0) on your computer, all of which are included with your C/C++ compiler. If you don't these files, you can go to step 10 now.

Choose Interpret | Make first to make sure the "user files" (those in c:\prj\usr) exist.

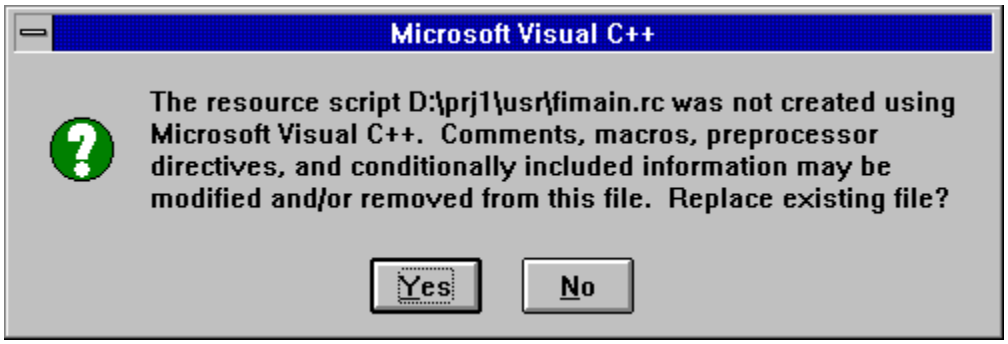
Go to your favourite resource editor (Resource Workshop for Borland users, VC++ IDE for MS users). Open c:\prj\usr\fimain.rc. If you use VC++, choose Resource | Set Includes and enter the parameters shown below:





Click on "OK". Then you will be warned:

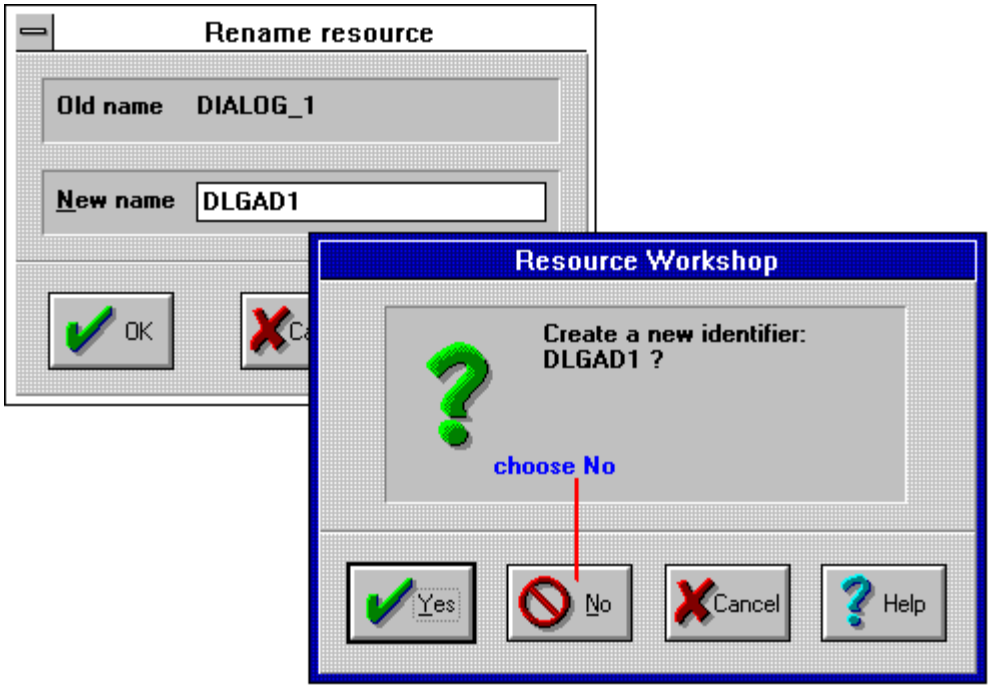
Simply click on "OK" to ignore it. Later when you save fimain.rc, you will be warned again:



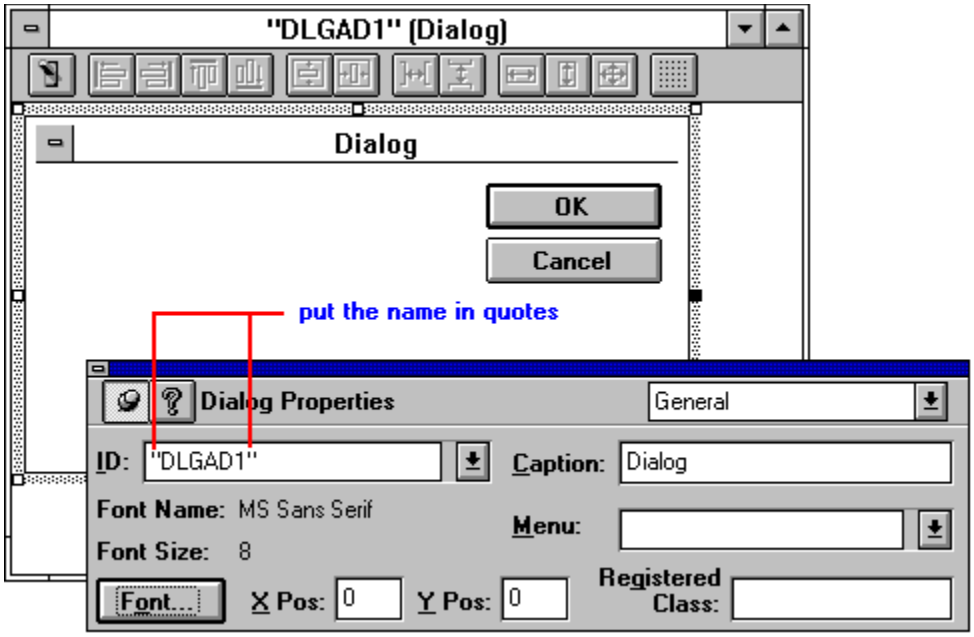
Click on "Yes" to get rid of it.

Note that the purpose of the steps above is to make sure the modified fimain.rc can still be compiled with rc.exe. They are only required if you're using VC++.

Then, create a dialog called "dlgad1". You must use ascii ID instead of integer ID. For example, in Resource Workshop:

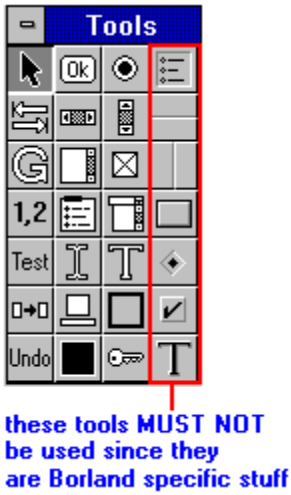


In AppStudio:



Turn off the "visible" flag. Otherwise you will experience flickering during the install.

Borland users must NOT use BWCC style dialogs here. This includes the steel chisel look background and the Borland controls:



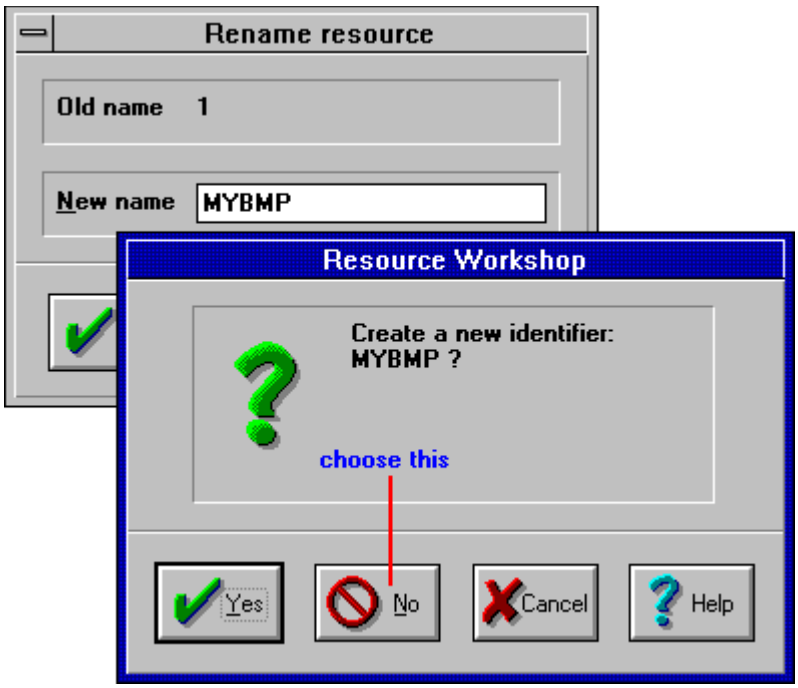
MS users will have to use some other resource editor to create a 256 color bitmap.

Step 7

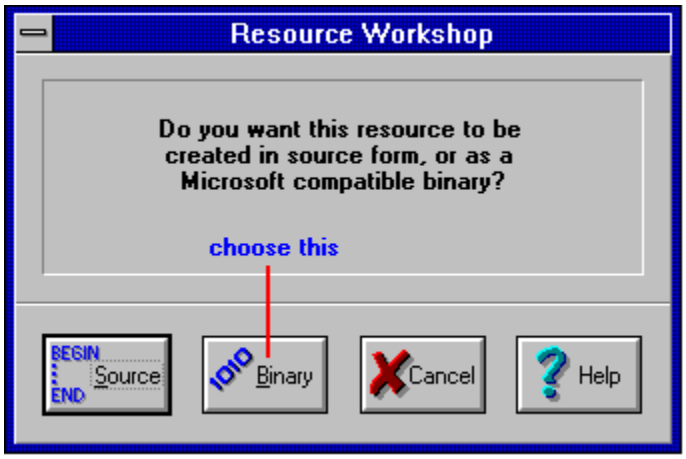
Create a 16 color (NOT 256!) bitmap called "MYBMP". This must be an ascii ID. This bitmap will be displayed when the user's display doesn't support 256 colors.

Create a 256 color bitmap called "MYBMP256". This must be an ascii ID. This bitmap will be displayed when the user's display supports 256 colors or above.

Let me repeat, the name of the bitmap MUST be ascii ID. For Borland users, after choosing Resource | Rename, you must tell Resource Workshop NOT to create a new ID:



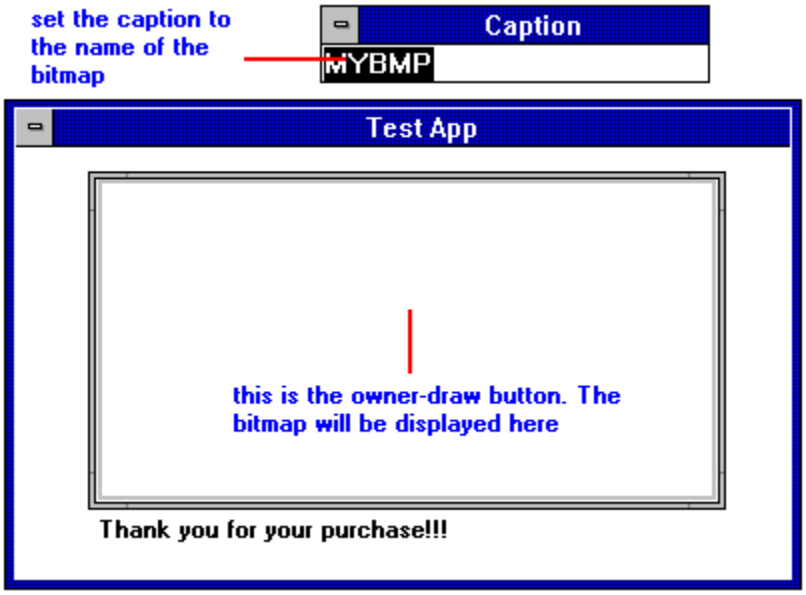
Borland users MUST use binary format as opposed to source format:



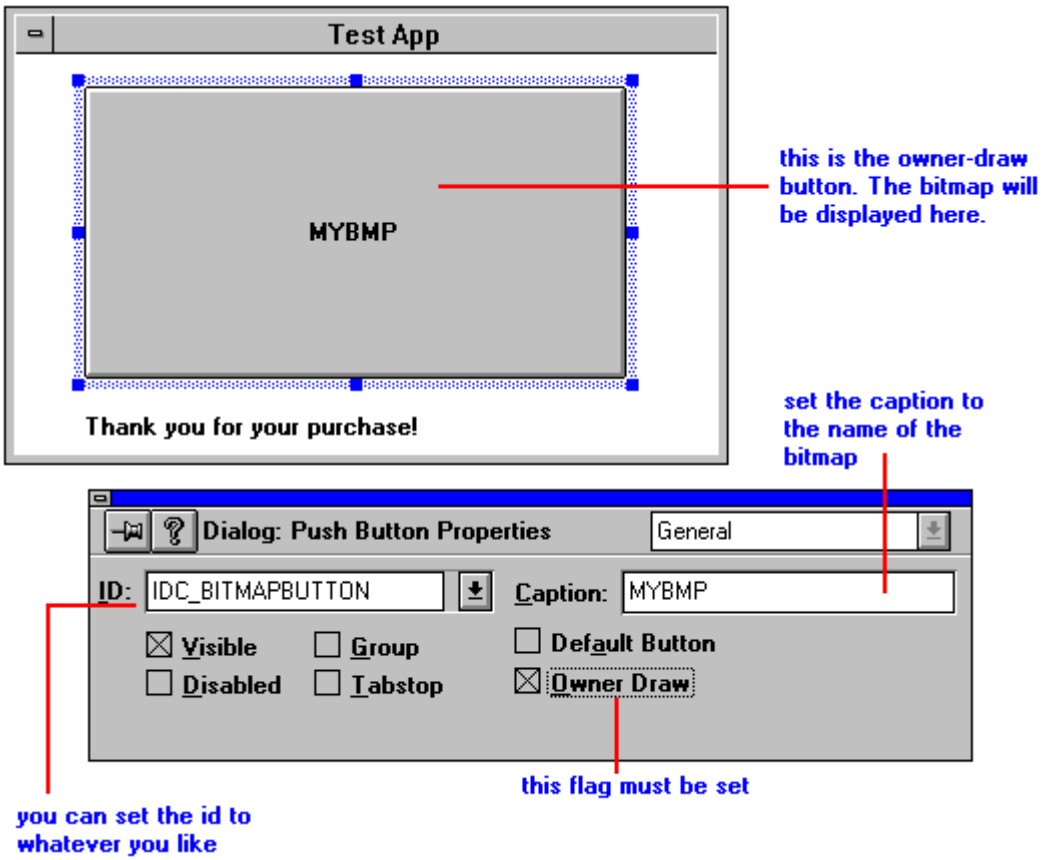
Step 8

Put an owner-draw button in the DLGAD1 and set its caption to "MYBMP".

For example, in Resource Workshop:



In AppStudio:



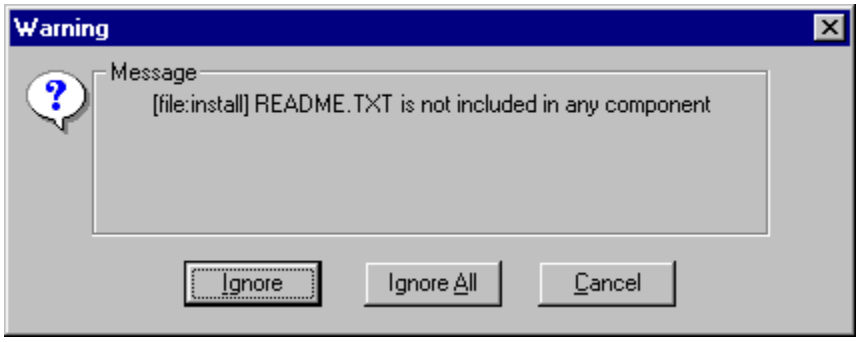
Save the resource file and go back to Freeman Constructor.

Step 9

Choose Edit | Advertising Dialogs. Add an entry for DLGAD1:

Click on "OK" button.

Step 10
Choose Interpret | Run to test run.



Oops! You didn't include readme.txt in any of your components and as a result it will never get selected and installed.

Click on "Cancel" button and then go to the component list (Edit | Software Components) to add readme.txt to all leaf components (remember? you can't include files in branch components), since you want to install and display readme.txt no matter what the user has selected.

Step 11
Now, try again (Interpret | Run) and build disk set, purify and zip the install later.

Congratulations! You have finished tutorial 3.