$1 #2 K3 ConGUI : A Simple Example Application

ConGUI - Is a simple example of how a standard 'console' application might also take advantage of some of the graphical capabilities of Windows NT.

There are many applications that for one reason or another would have nothing to gain by implementing full fledged 'GUI' interface with Title Bars, Menus, Scrollable Client Area etc. But that doesn't mean they have to be stuck in the world of characters forever...

Take the example of a compiler for instance.

By itself, and not as a 'module' of some monster IDE, a compiler really has no need for menus, button bars, or many of the other UI aspects of a standard Windows application. The user of a compiler simply wants to launch the compiler with special command line switches, and let it do its thing, then return control back to the user. In this scenerio however, there is a timer where the user might want the compiler to be 'friendlier' then what the current generation of compiler is. Do you know what all the command line switches are for your compiler? Do you occasionally want some on-line assistance with understanding what the compiler is capable of?

The current Microsoft C compiler, has a 'help' mode where you specify a command line of:

```
C:> CL /?
```

And it will promptly spill out 4 screens worth of information on the available switches. Missed one of them? Ooops, better run that command again, and this time have a pencil and paper ready to jot down some notes. Hmmm, was that a /c or /C for compile only? Ooops, better run that command again...

ConGUI illustrates the useage of a '/dialog' parameter, which will bring up a fully functional dialog box that will assist the user in selecting the parameters that they want for the command line.

Want to see a little more information about some of those switches? Then allow the user to access a full featured WinHelp file like other 'GUI' applications have. ConGUI shows how it is extrememly easy for a console application to make standard WinHelp calls to access the grapical help engine.

Another feature that ConGUI shows, is how it is possible for a console application to figure out how it was launched. It would often be useful for a text application to realize that it was being launched into its own window, and that when it closed down, the window it had written all of its output into would be closed down as well. Windows NT doesn't currently provide an easy mechanism for doing this, but by making a couple of calls to determine some of the window states, an application can do a pretty good job at figuring this out.

To see this in action, try the following methods of executing ConGUI, and notice how

it behaves:

- Double click on the .EXE from the File Manager
- Add it to a group in the Program Manager, and double click on it from there.
- From a command prompt, type : **`CONGUI`**
- From a command prompt, type : **`START CONGUI`**
- From a command prompt, type : **`CONGUI > TEMP.TXT`**

This sample application is not necessarily intended to demonstrate 'the' way that console applications should be written for Windows NT, it is mearly demonstrating a possible way you could add some additional features to your application to make it more useful to your users, and hopefully stand out from some of your competition.