

## **ResGauge v1.3**

23 August 1992

Richard Franklin Albury

PO Box 19652

Raleigh, NC 27619-9652

CompuServe ID: 76477,534

### **What is ResGauge?**

**ResGauge** is a simple, nonintrusive little utility which monitors free system resources in Windows 3.1. **ResGauge** started off as a simple utility about a year ago. I'd read some questions in the Windows forums on CompuServe about how to determine free system resources, I'd come across some information on the actual undocumented call used to do this, and I cobbled together a crude little utility in a few hours and uploaded it. Some people actually downloaded it and used it, and based on their suggestions, I took my crude little utility and smoothed some of the rough edges.

### **What are free system resources anyway?**

However much I may like some things about Windows - even though I started off on a Macintosh and I contracted for IBM on OS/2 development - it annoys the heck out of me that a system capable (in theory) of managing 64 MB of memory - 16 MB actual, 48 MB virtual - is held in check by two little 64 KB chunks of memory. Windows is essentially made up of three dynamic link libraries: the system module, the GDI (Graphics Device Interface) module, and the User (interface) module. The GDI and User DLLs each have a local heap, an area of memory in which they allocate memory for their own use. Since a local heap can't exceed 64 KB, each DLL is limited in the amount of memory it can allocate for its own use, giving rise to the notion of free system resources, usually expressed as the percentage of a DLL's local heap currently free. The number you see in the **About Program Manager...** dialog is the smaller of these two percentages, and this number will change as programs load and free menus, icons, and other interface resources. Not surprisingly, the Windows 3.0 Program Manager was one of the worst offenders in exhausting free system resources: every time you opened a group window, Program Manager would load the program icons but it wouldn't free them when the window was subsequently closed. This behavior has been fixed for Windows 3.1.

### **What's the catch?**

Version 1.3, like the versions before it, is distributed with the source and this document, and all I ask is that the package remain together. I placed version 1.0 in the public domain, but as with versions 1.1 and 1.2, I will ask for some compensation, sort of. Being an election year, and given that things don't seem to be getting much better - especially the environment! - I'm going to ask that if you use this utility and decide to keep it, you're obligated to get out and vote. I'm not going to let you off as lightly as it may appear, however: you're also going to have to give some careful thought to whom and what you vote for. ☺ Also, if you don't have access to CompuServe and all you have is this document, send me a check for \$5 and I'll send you a 3½" disk with the latest version, this document, and all the source code. Users outside the US can simply send me a letter with the smallest denomination of their country's paper currency or a couple of dozen canceled stamps (I have a daughter who loves such items).

## How does it work?

**ResGauge** only runs as an icon: to do things any other way didn't make much sense. The system menu has all the irrelevant items disabled - **R**estore, **S**ize, **M**inimize, and **M**aximize - and three new items added:

- **K**eep on top allows you to keep **ResGauge** "floating" above other windows, much like the Windows 3.1 **C**lock applet;
- **C**onfigure... allows you to modify **ResGauge**'s operation in some interesting and unique ways (this is described in more detail in the following section); and
- **A**bout... is the usual shameless blurb.

## What are all these settings for?

The **M**onitor section allows you to specify which of the two local heaps to monitor. You can also monitor both, and the number reported by **ResGauge** will be the smaller of the two. This is also the number reported by Program Manager, although I have noticed that **ResGauge** occasionally differed from Program Manager until I started using the "sanctioned" calls in Windows 3.1. One new feature for version 1.3 is that you can tell from the gauge label what **ResGauge** is monitoring: there's a **G** for GDI, a **U** for User, and a **B** for both in front of the percent free gauge caption. A second new feature for version 1.3 is the **G**auge **C**olor... button. If the default gauge colors - red for GDI, green for User, and blue for both - don't appeal to you, you can set them to whatever colors you like. The additional wrinkle with this is that you can set a color for each monitor state, such as red for GDI, blue for User, and green for both.

The **A**larm section allows you to set **ResGauge** to either beep or flash its icon when free system resources drop below the specified level. This is useful if you're more interested in getting work done than staring at my little program, so setting the alarm will give you a chance to save your work when system resources get low. If the beeping or the flashing annoy you, you can also turn this feature off. A change for version 1.3 is to bring **ResGauge** to the top when the **A**larm type is set to **F**lash but **K**eep on top is not enabled.

Finally, there are four buttons at the bottom of the **C**onfigure... dialog. **S**ave saves all the changes you've made to a private initialization file so that **ResGauge** will "remember" its settings the next time it runs. **A**pply lets you continue to run with the changes you've made without saving them. **D**efault resets all of **ResGauge**'s settings to default values, and **C**ancel undoes any changes you've made.

## How do I run three copies of ResGauge?

**ResGauge** will either read its configuration file when it starts or use default settings if the file doesn't exist. This scheme obviously doesn't lend itself to running different configurations simultaneously, so version 1.3 of **ResGauge** adds support for command line arguments: use **G**DI, **U**ser, or **B**oth to have **ResGauge** monitor the GDI heap, the User heap, or both heaps respectively. The case of the argument doesn't matter, but the spelling does, and an invalid argument will be ignored, as will more than one argument. Finally, if you used the **G**auge **C**olor... option in the **C**onfigure **ResGauge** dialog to set different colors for each mode, these will automatically be used.

## How do I install ResGauge?

The easiest way to install **ResGauge** is to have it run when you start Windows: simply add

**ResGauge** to the Startup group in Program Manager. If you want to run all three configurations as discussed in the previous section, create three **ResGauge** icons, each with a different argument.

### Who helped you?

I wrote this program by myself. Honestly! I did have some helpful suggestions from the following folks, though:

- **Anthony W. Rairden** for the **USER/GDI/Both** options and the alarm threshold idea
- **Greg Saddler** for the **Keep on top** option
- **Edward Bauman** for the color configuration idea
- **David Hoos** for the no-float/flash fix, some coding suggestions, and the correct spelling of "threshold." ☺
- **Larry LaBella** for displaying both USER and GDI data: I cheated and rewrote **ResGauge** so that you can run one copy for USER, one for GDI, and one for both, each with a different color if you want

### Aren't you done yet?

Almost. I'd like to say in closing that if you have any comments, kudos, complaints, or suggestions, I'd like to hear them. If you have any ideas for any other utilities you'd like to see, please let me know: I'd love to see them.

### Version history

- 1.0** initial version
- 1.1** added configuration code, changed display
- 1.2** converted to medium model - still under 10 KB!
- 1.3** converted to Windows 3.1 STRICT compliance, added Windows-3.1 compliant "floating," made gauge configuration more informative, added color configuration, removed the restriction to a single instance, added command line support, and fixed the no-float/flash situation - all this in less than 12 KB!