

MemWatch

For Microsoft Windows 3.1

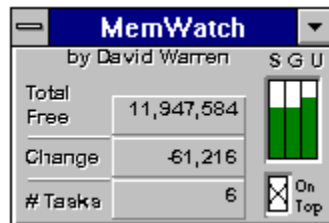
by David Warren

MemWatch is a simple utility, written in Microsoft Visual Basic that displays available system memory as well as real-time changes in memory use. It also includes a graphical display of System, GDI, and User resource demand, as well as a count of the number of tasks running. MemWatch will stay on top of all your other windows, so you can see at all times what changes are taking place.

I originally wrote MemWatch as an application prototyping tool so I could see, in real-time, the demands applications make on system memory and to gauge the effects of any optimizing modifications. It is also very useful for non-programmers who just want to see how Windows applications (and Windows, itself) make use of system memory.

To use MemWatch, just copy MEMWATCH.EXE to your hard disk. It doesn't need to be in any particular directory, although most users install it in their Windows or Windows\System directory. MemWatch requires Microsoft's Visual Basic runtime library, VBRUN200.DLL. If you already use programs written with Visual Basic 2.0, then you already have the DLL installed. If not, a copy is available from many bulletin boards or CompuServe. (As of this writing, VBRUN200.DLL mysteriously disappeared from both WINSHARE and MSBASIC forums on CompuServe -- certainly an accident that will be remedied by the time you read this!) VBRUN200.DLL should be installed in your Windows\System directory. I used a smaller than normal font in order to keep the window as small as possible. This happens to prevent MemWatch from running under Windows versions earlier than 3.1.

Just add MemWatch to any Program Manager group, or select Run from the File menu, type MEMWATCH and click OK. Be sure to include the full path if you installed MEMWATCH.EXE someplace other than your Windows directory. When MemWatch appears, you can move it to any convenient place on your desktop (it will remember it's new location and return there every time you start it. If you run out of desktop, run Memwatch as an icon. The MemWatch icon title will display the changing amount of memory.



The most interesting feature of MemWatch is the **Last Change** display. Whenever available memory changes, the amount of change will be displayed with either a minus (-) for memory used, or a plus (+) for memory released. For programmers, that means no more hopping between the ProgMan About box and the calculator. Which Visual Basic controls and VBX's use more memory? **Make an .EXE each way, and see the results immediately!**

It's also interesting to watch commercial applications use and release memory. Use MemWatch to see how a certain application uses different amounts of memory depending on the hardware configuration.

The resource gauge displays available percentages of resources for System, GDI and

User, labelled S, G and U, accordingly. The color bars will change as a further indication of the system status: from green to yellow at 50%, and yellow to red when a resource availability falls below 25%.

By default, MemWatch will stay on top of any other windows on your desktop. You can turn off the "stay on top" feature by clicking the On Top checkbox in the lower right corner of the MemWatch window. You can also have MemWatch start without staying on top by running:

MEMWATCH notop

What Visual Basic Programmers Can Expect To See

The first Visual Basic application also carries the some of VBRUN with it. Even if MemWatch is the first VB app to run, it can't measure itself. It can, however, measure the usage for subsequent instances of itself.

If MemWatch is the first VB app to run on your system, it will consume about 93K. It will then demonstrate that subsequent instances of MemWatch only use about 53K (even though the executable is under 9K in size). A similar situation arises with any VB application loaded before MemWatch, and other subsequent VB apps. Does that mean that the first VB app carries about 40K of VBRUN? Read the documentation for VB and then see if you can figure it out.

The smallest instance of a VB application I've seen is around 50K for a 4K formless utility that just runs out of a Sub Main() procedure (not counting MemWatch's 40K of VBRUN -- I assume you can add 40-50K to that if it happens to be the first VB app running). They can get into the megabytes. One very effective use of MemWatch is observing the effect of **Unload Form** instead of **Form.Hide**.

I wrote MemWatch in about an hour. It is free, subject to the following conditions:

1. It is used exclusively for non-commercial purposes. In non-commercial purposes, I include individual programmers who use it while making a living;
2. It is not included in whole or part with or within any other product, except shareware distribution disks, and, provided full credit is given to the author;
3. It is not transferred to another party in exchange for any form of payment or compensation, except reasonable material costs in the distribution of shareware collections;

and that if you like it, or have any comments whatsoever, you feel free to let me know.

MemWatch is distributed "as-is" and using the program constitutes your acceptance of these conditions and full responsibility for any and all damages that might arise as a consequence of its use. The author reserves all rights to any future modification or distribution of MemWatch.

MemWatch

for Microsoft Windows 3.1
by David Warren
June, 1993

David Warren
CompuServe: 72500,1406
Internet: davidw@cyberden.sf.ca.us
1642 Elda Court
Pleasant Hill, CA 94523-3012