



## **We Are Morons: a quick look at the Win2k source** **(Technology)**



By [Selznak](#)

Mon Feb 16th, 2004 at 07:38:15 AM EST

A quick, superficial look at the style and content of the leaked Windows 2000 source. I quote from the comments but not the code, so this should be safe for developers to read.

---

### **Overview**

Several days ago, two files containing Microsoft source code began circulating on the Internet. One contains a majority of the NT4 source code: this is not discussed here. The other contains a fraction of the Windows 2000 source code, reportedly about 15% of the total. This includes some networking code including winsock and inet; as well as some shell code. Some other familiar items include the event log, and some of the default screensavers.

The timestamps on the files generally say 25 July 2000. The source is contained in a Zip file of size 213,748,207 bytes, named windows\_2000\_source\_code.zip, which has been widely circulated on P2P networks. Some dummy files of similar size, containing just strings of zeroes, have also circulated.

There has been some speculation that while the bulk of the source is genuine, some of the comments have been tampered with to embarrass Microsoft. This is difficult to disprove, but I find it implausible. The embarrassing comments occur on thousands of lines, in realistic places. Furthermore, if someone had done that, it would have been easy to make the comments far more incriminating.

### **Embarrassments**

In the struggle to meet deadlines, I think pretty much all programmers have put in comments they might later regret, including swearwords and acerbic comments about other code or requirements. Also, any conscientious coder will put in prominent comments warning others about the trickier parts of the code. Comments like "UGLY TERRIBLE HACK" tend to indicate good code rather than bad: in bad code ugly terrible hacks are considered par for the course. It would therefore be both hypocritical and meaningless to go through the comments looking for embarrassments. But also fun, so let's go.

Curse words: there are a dozen or so "fucks" and "shits", and hundreds of "craps".  
Some dissatisfaction with the compiler is expressed in `private\shell\shell32\util.cpp`:

```
// the fucking alpha cpp compiler seems to fuck up the goddam type
"LPITEMIDLIST", so to work
// around the fucking peice of shit compiler we pass the last param as an
void *instead of a LPITEMIDLIST
```

Some insight into Microsoft's famous daily build process is given in  
`private\windows\media\avi\verinfo.16\verinfo.h`:

```
*
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
* !!!!!!!!!IF YOU CHANGE TABS
TO SPACES, YOU WILL BE KILLED!!!!!!!!!!
*          !!!!!!!!!!!!!!!!!!!!!DOING
SO FUCKS THE BUILD PROCESS!!!!!!!!!!!!!!!!!!!!!!
*
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

There are also various references to idiots and morons, some external, some within Microsoft. The file `private\ntos\rtl\heap.c`, which dates from 1989, tells us

```
// The specific idiot in this case is Office95, which likes
// to free a random pointer when you start Word95 from a desktop
// shortcut.
```

The file `private\ntos\w32\ntuser\kernel\swp.c` from 11-Jul-1991 points at

```
* for idiots like MS-Access 2.0 who SetWindowPos( SWP_BOZO
* and blow away themselves on the shell, then lets
* just ignore their plea to be removed from the tray
```

Morons also abound, as in this selection

```
private\genx\shell\inc\prsh.w:
// we are such morons. Wiz97 underwent a redesign between IE4 and
IE5
```

```
private\shell\ext\ftp\ftpdrop.cpp:
We have to do this only because Exchange is a moron.
```

```
private\shell\shdoc401\unicpp\desktop.cpp:
```

```
// We are morons. We changed the IDeskTray interface between IE4
```

```
private\shell\browseui\itbar.cpp:  
// should be fixed in the apps themselves. Morons!
```

Microsoft programmers also take their duty to warn others seriously. There are over 4,000 references to "hacks", mostly warnings. These include

```
private\inet\mshtml\src\core\cdbase\baseprop.cxx:  
// HACK! HACK! HACK! (MohanB) In order to fix #64710 at this very late
```

```
private\inet\mshtml\src\core\cdutil\genutil.cxx:  
// HACK HACK HACK. REMOVE THIS ONCE MARLETT IS AROUND
```

```
private\inet\mshtml\src\other\moniker\resprot.cxx:  
// <HACK>  
goto EndHack;  
// </HACK>
```

```
private\inet\mshtml\src\site\layout\flowlyt.cxx:  
// God, I hate this hack ...
```

```
private\inet\wininet\urlcache\cachecfg.cxx:  
// Dumb hack for back compat. *sigh*
```

```
private\inet\wininet\urlcache\filemgr.cxx:  
// ACHTUNG!!! this is a special hack for IBM antivirus software
```

```
private\ispu\pkitrust\trustui\acuictl.cpp:  
// HACK ALERT, believe it or not there is no way to get the height of the  
current  
// HACK ON TOP OF HACK ALERT,
```

```
private\ntos\udfs\devctrl.c:  
// Add the hack-o-ramma to fix formats.
```

```
private\shell\shdoc401\unicpp\sendto.cpp:  
// Mondo hackitude-o-rama.
```

```
private\ntos\w32\ntcon\server\link.c:  
// HUGE, HUGE hack-o-rama to get NTSD started on this process!
```

```
private\ntos\w32\ntuser\client\dlgmgr.c:  
// HACK OF DEATH:
```

```
private\shell\lib\util.cpp:
```

```
// TERRIBLE HORRIBLE NO GOOD VERY BAD HACK
```

```
private\ntos\w32\ntuser\client\nt6\user.h:
```

```
* The magnitude of this hack compares favorably with that of the  
national debt.
```

While surprisingly informal, there are limits to how far the programmers go. There are no derogatory references to Microsoft or Windows themselves. Bill Gates is never mentioned. There are no racist or homophobic slurs. I saw only one drug reference.

```
private\shell\ext\tweakui\genthunk.c:
```

```
* CallProc32W is insane. It's a variadic function that uses  
* the pascal calling convention. (It probably makes more sense  
* when you're stoned.)
```

## Quality

Despite the above, the quality of the code is generally excellent. Modules are small, and procedures generally fit on a single screen. The commenting is very detailed about intentions, but doesn't fall into "add one to i" redundancy.

There is some variety in the commenting style. Sometimes blocks use a // at every line, sometimes the /\* \*/ style. In some modules functions have a history, some do not. Some functions describe their variables in a comment block, some don't. Microsoft appears not to have fallen into the trap of enforcing over-rigid standards or universal use of over-complicated automatic tools. They seem to trust their developers to comment well, and they do.

However, not everything is so rosy. Some of the modules are clearly suffering from the hacks upon hacks mentioned earlier. As someone who struggled immensely trying to get the MSInet control working not long after this code was released, it's a relief to see that the inet code is as bad as I thought.

From the comments, it also appears that most of the uglier hacks are due to compatibility issues: either backward-compatibility, hardware compatibility or issues caused by particular software. Microsoft's vast compatibility strengths have clearly come at a cost, both in developer-sweat and the elegance (and hence stability and maintainability) of the code.

## Open Source

It's been widely rumored for a while that Microsoft relies on stolen open source code. The rumor has faced widespread skepticism too. Microsoft has hundreds of millions of lines of code, most of it highly specialized. Hardly any of that could benefit from stealing: it hardly seems worth the legal risk. It's true that early versions of the TCP-IP stack were (legally) taken from BSD: but that was a long time ago, when Microsoft was much smaller.

Searching the code for "linux" and "GPL" finds no references. "BSD" finds only a

couple of references to BSD-convention strings. "GNU" finds a lot of references to a GNUmakefile in private\genx\shell, which in turn mentions a "mode for Emacs." This is apparently legitimate: simply using a makefile does not apply the makefile's copyright to the resulting code.

Therefore, a superficial look at the code finds no evidence that Microsoft has violated the GPL or stolen other open source code. Closer examination might turn something up.

### **Favoritism**

It's noticeable that a lot of the "hacks" refer to individual applications. In some cases they are non-Microsoft, such as [this case](#): a Borland compiler came to depend on an existing bug, so their fix worked to preserve some of the bug's behaviour. But just as often these application-specific fixes are for Microsoft's own apps. There seems to be an informal hierarchy when it comes these: Microsoft apps take precedence, then major software companies like IBM and Borland.

It's also interesting to finally see references to the notorious undocumented features, which Microsoft application developers have long been known to use.

```
private\mvdm\wow32\wcntl32.c:  
// These undocumented messages are used by Excel 5.0
```

```
private\mvdm\wow32\wgdi31.c:  
// InquireVisRgn is an undocumented Win 3.1 API. This code has been  
// suggested by ChuckWh. If this does not fix the s 2.0  
// problem, then ChuckWh would be providing us with an private entry  
// point.
```

```
private\mvdm\wow32\wgfont.c:  
* This thunk implements the undocumented Win3.0 and Win3.1 API  
* GetCurLogFont (GDI.411). Symantec QA4.0 uses it.  
* To implement this undocumented API we will use the NT  
undocumented API
```

In some cases, the programmers themselves appear to have been frustrated or surprised.

```
private\ntos\w32\ntuser\kernel\mnpopup.c:  
// Set the GlobalPopupMenu variable so that EndMenu works for  
popupmenus so  
// that WinWart II people can continue to abuse undocumented  
functions.
```

```
private\windows\shell\accesory\hypertrm\emu\minitel.c:  
// Guess what? Latent background color is always adopted for mosaics.  
// This is a major undocumented find...
```

```
private\windows\shell\accesory\hypertrm\emu\minitelf.c:  
// Ah, the life of the undocumented. The documentation says  
// that this guys does not validate, colors, act as a delimiter  
// and fills with spaces. Wrong. It does validate the color.  
// As such its a delimiter. If...
```

## Conclusions

The security risks from this code appear to be low. Microsoft do appear to be checking for buffer overruns in the obvious places. The amount of networking code here is small enough for Microsoft to easily check for any vulnerabilities that might be revealed: it's the big applications that pose more of a risk. This code is also nearly four years old: any obvious problems should be patched by now.

Microsoft's fears that this code will be pirated by its competitors also seem largely unfounded. With application code this would be a risk, but it's hard to see Microsoft's operating system competitors taking advantage of it. Neither Apple nor Linux are in a much of position to steal code and get away with it, even if it was useful to them.

In short, there is nothing really surprising in this leak. Microsoft does not steal open-source code. Their older code is flaky, their modern code excellent. Their programmers are skilled and enthusiastic. Problems are generally due to a trade-off of current quality against vast hardware, software and backward compatibility.

Full discussion: <http://www.kuro5hin.org/story/2004/2/15/71552/7795>

---

All trademarks and copyrights on this page are owned by their respective companies. The Rest © 2000 - 2001 Kuro5hin.org Inc.