

**Erik Stevenson**

---

**From:** Brad Silverberg  
**To:** Mike Maples  
**Cc:** Cameron Myhrvold; Lewis Levin; Paul Maritz; Pete Higgins; Claire Lematta  
**Subject:** RE: Undoc api mail  
**Date:** Friday, August 28, 1992 4:06PM

internal communications routines should be kept private. every os has them. they will change from version to version. at the same time we need to provide reasonable hooks for debuggers and other systems tools. we have a good start with toolhelp.dll that's in 3.1.

just like undoc api's there are also undoc messages that the system uses internally.

most of the shell api's we have described in one way or another, such as the recent msj article on how to do drag/drop sources.

some parts of the system are not meant to be replaced (like a rasterizer). it may be possible thru hacking internal data structures, but we should don't make it easy and don't document how. there are aspects of sparta that are the same, like sharing a directory or printer. internal data structures in the server need to be hacked. dirty stuff.

-----  
**From:** Mike Maples  
**To:** bradsi  
**Cc:** cameronm; lewisl; paulma; peteh; w-clairl  
**Subject:** RE: Undoc api mail  
**Date:** Friday, August 28, 1992 3:06PM

What is the problem with documenting all of them?

-----  
**From:** Brad Silverberg  
**To:** Cameron Myhrvold; Lewis Levin; Paul Maritz; Pete Higgins; Claire Lematta  
**Cc:** Mike Maples  
**Subject:** FW: Undoc api mail  
**Date:** Fri, Aug 28, 1992 12:50PM

-----  
**From:** David D'Souza  
**To:** bradsi  
**Subject:** RE: Undoc api mail  
**Date:** Friday, August 28, 1992 11:32AM

The main undocumented apis 3rd party people are using are those we put in for the shell (Winfile, progman, control panel, printman). These include things like DragDrop, a couple new listbox messages for winfile, icon extraction from non-running exes, WinNet apis, and the tiling/cascade code used by Taskman. Other than drag/drop, these were new undocumented apis we added at the last moment to simply allow us to enhance the shell in various ways. These are predominantly the apis third party apps tend to call. These are also the apis we cleaned up and documented for 3.1. For example, we added a real ExtractIcon api to allow people to grab icons from apps; the old 3.0 method using undocumented had gp faulting bugs which we found after ship.

Another very large source for "undocumented" apis are those that

provide internal communication between CORE components and for communication between CORE and device drivers. For example, USER calls private apis in KERNEL to retrieve private datastructures such as the message queue for the current task. The message queue is a private struct which apps don't need to manipulate so we don't document this structure or the method for accessing it. We also needed the ability to extend this structure for things such as Pen Win support in 3.1 and 32 bit window handles for Chicago so it was to our advantage to keep it private. But turns out, it is also useful for debuggers to be able to dump the apps message queue so some tools use private apis to access this. An example of CORE/Device driver communication are entry points for the mouse driver/keyboard driver to call into user. These are undocumented in the SDK and are often glossed upon or unclearly documented in the DDK.

Finally there are some internal routines that became callable as apis due to the real mode days. To handle the movable memory before selectors, many more functions had to be exported so that Kernel would properly patch our movable ds to the correct value. Unfortunately, this also means the apis are callable by external apis but, at the time, we weren't concerned about this and simply left these entry points undocumented. These internal routines were often helper routines to more specific documented apis. But it turned out some older apps, such as Excel, called some of these routines from Win2.x days. Along the same lines, there were some Windows 1.x and Windows 2.x apis which we retired since new concepts have replaced them. For example, many apis dealing with tiled windows have simply been removed from the documentation and retired.

I can only think of one instance where we added functions for ISVs but the function remained undocumented. Back in the Windows 2.x days, some functions were added for IRIS but in 3.0, the apis were either removed or documented. For all other ISV requests (either 3rd party or MS), we added documented APIs to the next version of the product and in some rare cases, provided them with an undocumented means to do what they wanted in the previous version.

-Dave