

Re: Is it worth upgrading to XP Pro ?

Source:

<http://www.tech-archive.net/Archive/WinXP/microsoft.public.windowsxp.basics/2004-11/1912.html>

From: FACE (*AFaceInTheCrowd_at_bellsouth.net*)

Date: 11/17/04

Date: Tue, 16 Nov 2004 21:29:32 -0500

On Tue, 16 Nov 2004 23:02:18 +0000 (UTC), "Kevin Lawton"
<socks.kepla.shoes@btinternet.com> in microsoft.public.windowsxp.basics
wrote:

>FACE <AFaceInTheCrowd@bellsouth.net> wrote:
> <snip>
>| My understanding is that Windows product up until XP allocated two 64K
>| segments for User/GDI resources. I am of the opinion that the glitzy,
>| flashy, internet of today eats the SR ravenously and the result is a
>| machine lockup on GDI failure. My further understanding is that XP
>| dynamically allocates further resource segments on an "as needed"
>| basis.
> <snip>
>I would have expected that the allocations to User/DGI resources were
>specified by values in the registry – let's face, just about everything else
>is !
>Could anyone possibly confirm this and, it is so, suggest what those
>registry values might be ?
>I tend to run my systems with lots of RAM – 512 Mb minimum, and up to 1 Gb –
>so 64Kb is a pathetic amount to allocate to something which is easily filled
>to the point of causing a problem.
>Kevin.

>

>

Kevin,

Yes, it is surprising.

I can relay the in-depth answer that I got on
microsoft.public.win98.performance, back in August. The thread name was
"Increase USER and GDI resources?" and it may shed light on this holdover
from the halcyon days of no viruses and no spyware:

~~~~~

Not to worry, I'm only rude in response to ill manners. :-)

The resource pools and their 64k limit are a gift of the compatibility gods.  
Windows 3.1 was a 16-bit operating system, so if you do the math (2<sup>16</sup>) you

microsoft.public.windowsxp.basics: Re: Is it worth upgrading to XP Pro ?

get 65,536 (or 64kb) as the maximum size that a memory pool can be. When Windows 95 came out it used a 32-bit memory model but it needed to support those older 16-bit programs, so it maintained the User and GDI pool sizes so they'd run correctly.

At about the same time as Windows 95 came Windows NT. The NT kernel attempted to handle these older 16-bit programs by running them in a virtual session – carve out a chunk of memory and make it look like a 16-bit system, then load and run the program in that chunk of memory. The problem was (and still is!) that this breaks as many programs under Windows NT as it fixes.

The sad part is that only the User and GDI pools are limited in Win9x – there are other 32-bit pools that can be used. And you can dynamically destroy items you've placed in the User and GDI pools when you're done with them, freeing up that memory for other uses. So, why don't they? I don't know. Maybe programmers are fundamentally lazy and use the User and GDI pools the way they do because it's easier.

(n.b. – I am a programmer and I am lazy, as are many of my programmer-friends, but I don't assume this tendency transfers to all other programmers.<G>)

But to get back on point ... if you could change the size of the User and GDI pools you would break all sorts of interesting things when a program assumes they'll be the correct size and dips into them to pull out a resource. So you'd also have to modify programs to expect a larger than expected pool. Neither of these are trivial tasks and would run the risk of breaking operating system functions that expect the User and GDI pools to be 64kb in size. So you'd have to modify Windows as well.

--

Richard G. Harper [MVP Win9x] rgharper@email.com  
\* PLEASE post all messages and replies in the newsgroups  
\* for the benefit of all. Private mail is usually not replied to.  
\* HELP us help YOU ... <http://www.dts-l.org/goodpost.htm>

~~~~

I don't think Richard would mind me reproducing this...it is in the google archives anyway.

Anyway, ultimately that was the answer i got as to if i could increase USER/GDI under Win 98 after i had poked a little bit (apparently not rudely though :-)) for an answer greater in detail than "No."

FACE