

What are the consequences of increasing the per-process GDI handle limit?

devblogs.microsoft.com/oldnewthing/20210831-00

August 31, 2021



Raymond Chen

A customer had a line-of-business application that frequently bumped into the default limit of 10,000 GDI handles per process. The wanted to know if it was safe to raise the limit.

Increasing the per-process GDI handle limit means that a single process can consume a larger percentage of GDI resources, starving out other processes. The current value was chosen to mitigate the impact of a process that consumes a large amount of GDI resources. That process could be consuming a large amount of GDI resources maliciously (trying to trigger a denial of service by exhausting system resources), or it could be inadvertent (a bug in the app causes it to leak GDI resources).

Preventing a process from consuming all of the GDI resources makes it more likely that there will be enough GDI resources remaining to do things like open a command prompt or Task Manager to terminate the runaway process.

In this customers case, the high usage is due to the app simply having a large GDI object requirement in its natural course of business. You can raise the limit in order to accommodate your program. Just be aware that you are also raising the limit for the other two types of programs (malicious or inadvertent resource consumption) and opening yourself to risk of attack from those types of apps.

In the meantime, you might want to see what you can do to get your program to consume fewer GDI resources in its natural course of business. That would let you return the handle limit to its normal value and simplify deployment.

[Raymond Chen](#)

Follow

