

Re: Trying to use MemoryMappings but having a simple problem...

Source: <http://coding.derkeiler.com/Archive/Delphi/comp.lang.pascal.delphi.misc/2005-01/0080.html>

From: Rob Kennedy (*me3_at_privacy.net*)

Date: 01/05/05

Date: Wed, 05 Jan 2005 10:30:39 -0600

jklimek@gmail.com wrote:

- > 1) Will each process that attaches to the DLL and then gets a
- > MappingHandle (to the shared memory aka file mapping) have the same
- > handle? (eg. will `IntToStr(MappingHandle)`) produce the same integer?

Probably not. That is, the handles `_might_` have the same value, but you should consider it nothing more than a coincidence when it happens. By the way, the same applies to the hook handle.

To pass a handle to another process, you need to use the `DuplicateHandle` API function. It allows you to generate a handle that will be valid in another process. (It won't be valid in the process that creates it.)

But you don't really need to do that for mapping handles since each interested process can simply open the file mapping by name. Be sure to read the documentation carefully about choosing unique names accessible to everything that's supposed to have access to it. The rules differ by OS version.

- > 2) Same as the above, but will each pointer (eg. `HookInformation =`
- > `MapViewOfFile()`) have the same address?

Nope.

You can `_ask_` for the file to be mapped to a particular address, but there's no guarantee that that's the address you'll get. It all depends on what addresses are already reserved.

When you want to send another program an address in the memory-mapped file, you should instead send just the offset into the file -- pass the address relative to the file's base address returned from `MapViewOfFile`. The other process can then add that offset to the address `MapViewOfFile` returns in `_that_` process. Memory-mapped files allow you to share memory, but not address space. Each process's address space is entirely its own.

--

comp.lang.pascal.delphi.misc: Re: Trying to use MemoryMappings but having a simple problem...

Rob