

# How do I force RunOnce commands to run in a specific order?

 devblogs.microsoft.com/oldnewthing/20250515-00/?p=111181

May 15, 2025



A corporate customer wanted to deploy Windows images with custom RunOnce entries that had to run in a specific order. The documentation says that if there is more than one entry under the RunOnce key, the order in which those programs run is “[indeterminate](#).” The customer did some experimentation and found that in practice the order comes from [algorithm redacted]. They planned on using this information to reverse-engineer the precise way to create their RunOnce keys so that after passing through the algorithm, they come out in the desired order.

Please don't do that.

The documentation says that the order is indeterminate. This means that the ordering is not contractual and could change at any time. The so-called algorithm is not something that was consciously designed. It is an emergent behavior that is a consequence of the interaction of internal implementation details of the registry and the shell.<sup>1</sup>

Code that takes a dependency on these subtle implementation details are always a thorn in the side of the application compatibility team. The registry team might change the way they record and enumerate subkeys to be more efficient, but doing so inadvertently breaks people who had taken a dependency on the old inefficient enumeration. And it will probably take a lot of work to figure out precisely what change to the system broke this corporate deployment, since it's not like there's a clear linkage between “registry performance optimization” and “corporate deployment fails.”

What will probably happen is that the corporate deployment failure won't be detected by application compatibility testing (because the corporate deployment is not something that is in the application compatibility database), and then the company will realize that “Oh no, after the most recent update, our deployments stopped working.” This will probably happen long after any notes on hidden deployment dependencies have been lost (if they even existed at all), and whoever designed the deployment images has since left the company. Now the company might be forced to pay for an expensive outside contractor to figure out the mess because their deployment scripts don't work on newer releases.

The solution is to work within the confines of the existing contract.

You can create a single RunOnce entry that in turn runs each of the tasks in a specific order. That way, it doesn't matter what order the system chooses for the RunOnce entries. Once it gets to yours, your programs will all run in sequence in the order you specify.

**Related reading:** [What order do programs in the startup group execute?](#)

<sup>1</sup> Windows 95 and Windows NT enumerate registry values differently, so your reverse-engineered algorithm wouldn't have worked on Windows 95. That's once nice thing about the era when Windows 95 and Windows NT overlapped: There were two largely independent implementations of Win32, so things that were documented as "unspecified" frequently did behave differently between the two code bases, and since people cared about supporting both families of operating systems, they didn't take dependencies on these unspecified implementation-dependent details.