

# Why is the StartRunNOHOMEPATH policy so very specific about what it does?

[devblogs.microsoft.com/oldnewthing/20151118-00](http://devblogs.microsoft.com/oldnewthing/20151118-00)

November 18, 2015



Raymond Chen

The StartRunNOHOMEPATH policy affects whether the user's HOME path is the current directory when you run a program from the Run dialog. But that's all it does. DWalker asks, "I wonder why the behavior is different between Start and Start.Run."

The Start.Run dialog is not related to the Start menu. It's just some dialog box that happens to be directly invocable from Start. This dialog is also used by Task Manager when you say File.New Task, and you can also invoke it programmatically.

The search box on the Start menu is a search box, not a run box. Its job is to use what you typed as the basis for a search and present you a list of matches, and to execute the top match if you press Enter. That it runs Notepad if you type N-O-T-E-P-A-D is a consequence of the search rather than any attempt to run the command line `"notepad"`.

As for why the policy is so narrowly-defined, a lot of that falls back to the customer who requested the policy in the first place. Many policies are the result of direct requests from IT departments who want to control some very precise aspect of the system because it affects their network or otherwise creates a burden on their computing infrastructure. You can usually spot these types of policies because of their extremely precise formulation.

morlanweb wonders if there are equivalent policies for other launch points. Given that the `StartRunNoHOMEPATH` policy was created in response to a specific customer request, it stands to reason that corresponding policies for other launch points would exist only if a customer requested them. And so far, no corporate IT department has asked for them.

As a rule, IT departments do not notify us when they no longer need a policy, so these random strange-looking policies can linger for a long time without any real clients. Not that it would help, because once the policy is published, any other company's IT department could start using it without telling us.

These special one-off policies are tested primarily by the customers who requested them. If the policy implementation solves their problem, then that's that. This helps to shift the support burden of the policy from the product team to the individual who requested the policy. If a service pack or a future version of Windows causes the policy to behave strangely, we rely partly on the customer letting us know. In a sense, the loser become responsible for testing his check box.

Raymond Chen

**Follow**

