

# There are only twelve function keys, and who says there's somebody there to push them?

[devblogs.microsoft.com/oldnewthing/20080610-00](http://devblogs.microsoft.com/oldnewthing/20080610-00)

June 10, 2008



Raymond Chen

One of the proposals for adding hidden features to Windows Setup is to have a bunch of hidden function keys, one for each hidden option. Well, first of all, there are only twelve function keys and there are way more than twelve possible Setup features. (Yes, there may be a few function keys still active during Setup, but they exist only for compatibility purposes. No new Setup features involve function keys as far as I'm aware. And yes, there are keyboard layouts with more than twelve function keys, but think about what you're saying in context: "Yes, Setup should use function keys that are not available on most keyboards.") Furthermore, there's no guarantee that there's somebody sitting in front of the computer when it's running Setup. Indeed, 90% of the time, there is *nobody there at all*; Setup is running on a factory floor somewhere churning out computers day and night. Adding a command line option to Setup also runs you into problems: The command line gradually gets bloated with a bajillion options. After taking your several dozen customizations and cramming them onto the command line, you'll find yourself having to type 500 characters onto the command line to get what you want, and woe unto you if you typo one of those 500 characters. Can you imagine if somebody said, "To set up Windows the way I like, I simply type the following command," followed by a command that goes on for six lines? "This is total crap. I'm expected to type this monstrosity?" Windows Setup for quite a long time has supported so-called *unattended installation*. You build a so-called *unattend file* and pass it to the setup program as a single command line parameter, something like `setup /unattend:unattendfile`. (I forget the command line exactly; you can go look it up yourself.) Now somebody can post their favorite settings onto their Web site, and you can download it and pass it to Setup.

For many years, this unattend file took the form of an INI file, with `[Sections]` and `Key=Value` entries. In Windows Vista, the Setup folks threw away the old parser and switched to XML. Because XML is the wave of the future, right? (I can imagine Steve Ballmer jumping up and down shouting "XML! XML! XML!") I don't know whether they've actually done it, but in principle, the switch to XML means that they can write a schema for the unattend file, and then all the standard XML validation tools become available.

Raymond Chen

**Follow**

