How do you come up with new shortcut keys?

devblogs.microsoft.com/oldnewthing/20121022-00

October 22, 2012

Raymond Chen

Anon asks, "How do you come up with new shortcut keys and how do you deal with different keyboard layouts? What is the process; is there a company-wide procedure to keep things consistent?" This is several questions (none of them really a suggestion, but I've given up on making the Suggestion Box about suggestions; now it's just the "Ask Raymond a question" page), so let's take them one at a time. (Note that if you ask multiple questions, you reduce the likelihood that I'll answer them, because I feel obligated either to answer all of them or none of them.) First question: How do you come up with new shortcut keys? You just make them up. Application shortcut keys are local in scope (either to a dialog box, or to a menu, or to a top-level window message loop via the accelerator table), so you can just pick something that does not cause duplication within your local sphere. Second question: How do you deal with different keyboard layouts? To deal with different keyboard layouts, you, um, deal with different keyboard layouts. Menu shortcuts, dialog box shortcuts, and accelerator tables are all localizable, so translators can assign them to whatever key they feel works best on the keyboard layouts most likely to be used by their specific audience. If you want your shortcut to be invariant across keyboard layouts (such as `Ctrl`+`C` for copy, regardless of language) then you (or somebody you trust) need to check all the keyboard layouts to make sure your desired shortcut key is available on all of them. For example, the keyboard shortcuts for snapping, unsnapping, and repositioning applications in Windows 8 were originally going to be `Win`+`[` and `Win`+`]`. However, upon checking with the globalization folks, it became clear that those hotkeys were not going to work, because on many keyboards, typing `[` and `]` requires the use of the `AltGr` modifier. Users would have had to press `Win`+`AltGr`+`[` to move the gutter to the left, or if they didn't have an `AltGr` key, `Win`+`Ctrl`+`Alt`+`[`. This was clearly far too unwieldy, so the hotkeys were changed to `Win`+`.` and `Win`+`Shift`+`.`. Third question: Is there a company-wide procedure to keep things consistent? Since application shortcut keys are local in scope, there is no need for a company-wide procedure to keep them from conflicting with each other. Each application can assign its shortcut keys as it sees fit. And maintaining consistency across applications is done the same way gas stations and airlines maintain consistency in pricing: You look at what everybody else is doing and try to be consistent with them. This creates a first-move advantage: If the first application to create a Properties menu item assigns it the keyboard shortcut `Alt`+`R`, then that makes it likely that the second and third applications which

add a Properties menu item are likely to use the same shortcut. Windows 95 introduced a lot of new items into context menus (such as Properties), so the Windows 95 design team got to choose a lot of important keyboard shortcuts. In Explorer, hit `Shift`+`F10` and look at the Properties menu item. On English systems, it shows up as Properties. The shortcut is *still* `Alt`+`R`, over a decade later. Besides, imagine if there was some company-wide procedure for keeping things consistent. First, you would have to make sure everybody knew about this procedure. "What, there's a procedure I have to follow before I can assign menu and dialog shortcut keys and application accelerators?" And then everybody would complain, "Oh great, before I choose a shortcut, I have to fill out this application in quadruplicate and then show up at *yet another meeting*." Microsoft is famous for its burdensome meetings. You don't address this problem by adding more meetings. (Note that I don't share that author's views, probably because I don't go to many meetings.)

New shortcut keys are added all the time. A single dialog box can have a dozen or more. Imagine being on the shortcut-key-consistency committee. Your day would be wall-to-wall meetings about shortcuts. "Yes, the standard shortcut for Properties in German is `Alt`+`E` (for *Eigenschaften*), but we'd like to apply for an exception because our menu also has an Insert command (German: *Einfügen*) and the standard shortcut for that is also `Alt`+`E`. Since Insert is used much more often, we'd like to keep `Alt`+`E` for *Einfügen* and use `Alt`+`G` for *Eigenschaften*." And when somebody came to you with the request, how would you decide whether to approve or reject it? Probably by saying, "Well, let's see how other applications handle this." So your big complicated hotkey approval process reduced to what people would have done anyway. Congratulations, you just wasted everybody's time.

Raymond Chen

Follow