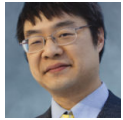


Windows has supported multiple UI languages for over a decade, but nobody knew it

devblogs.microsoft.com/oldnewthing/20110711-00

July 11, 2011



Raymond Chen

In the early days of Windows, there was a separate version of Windows for each language, and once you decided to install, say, the French version of Windows, you were locked into using French. You couldn't change your mind and, say, switch to German. The reason for this is that there were bits and pieces of language-dependent information stored all over the system. One obvious place is in file names. For example, a shortcut to the calculator program was kept at `%USERPROFILE%\Start Menu\Programs\Accessories\Calculator.lnk` on US-English systems, but `%USERPROFILE%\Startmenü\Programme\Zubehör\Rechner.lnk` on German systems. The name of the physical file system directory or file was displayed to the user as the name of the menu item. This means that if you started with an English system and simply replaced all the user interface resources with the corresponding German ones, you would still see a folder named *Accessories* on your Start menu, containing a shortcut named *Calculator*, even though they should now be displayed as *Zubehör* and *Rechner*. The registry was another place where language-dependent strings were stored. For example, file type descriptions were stored in plain text, which meant that if you installed an English system, then `HKEY_CLASSES_ROOT\txtfile` had the value `Text Document`, and that's the value shown to the user under the *Typ* column even though the user had switched the user interface resources to German. For Windows 2000, an effort was made to move all language-dependent content into resources so that they could be changed dynamically. If you need to store a language-dependent string anywhere, you can't store the string in plain text, because that would not survive a change in language. You have to store an indirect string and convert the indirect string to a real string at runtime, so that it mapped through the user's current user interface language. It was quite an effort identifying all the places that needed to be changed to conform to the new rules while still ensuring that the new rules were backward compatible with old code that followed the old rules. For example, you couldn't just say "To register a language-aware file type friendly name, write an indirect string to `HKEY_CLASSES_ROOT\progid`. For example, set `HKEY_CLASSES_ROOT\txtfile` to `REG_SZ:@C:\Windows\system32\notepad.exe,-469`." If you did that, then applications

which retrieved file type friendly names by reading directly from

`HKEY_CLASSES_ROOT\progid` (instead of using functions like `SHGetFileInfo`) would end up showing this to the user:

Name	Type	Modified
House pictures	@C:\Windows\system32\zipfldr.dll,-10195	11/16/1998 4:09 PM
notes	@C:\Windows\system32\notepad.exe,-469	11/23/1998 1:52 PM
Proposal	@C:\Program Files\Windows NT\Accessories\WORDPAD.EXE",-190	10/31/1998 10:32 AM

instead of

Name	Type	Modified
House pictures	Compressed Folder	11/16/1998 4:09 PM
notes	Text Document	11/23/1998 1:52 PM
Proposal	Rich Text Document	10/31/1998 10:32 AM

Designing and implementing all this was a major undertaking (that's what happens when you have to retrofit something as opposed to designing it in from the beginning), and to keep the test matrix from growing quadratically in the number of supported languages, a decision was made early on to support dynamic language changes only if the starting language is English. So yes, you could have both English and Dutch resources installed, but you have to start with English and add Dutch and not the other way around. Mind you, the implementation in Windows 2000 was not perfect. There were still places where English strings appeared even after you switched the user interface language to Dutch or German, but things got better at each new version of Windows. Unfortunately, pretty much nobody knew about this feature, since it was marketed to large multinational corporations and not to your random everyday users who simply want to change the user interface to a language they are more comfortable with. For Windows 2000 and Windows XP, you still had two ways of installing Windows with a German user interface: You could either install the English version and then add the German language pack (the fancy Windows 2000 multilingual way), or you could install the fully-localized German version of Windows, just as you always did. In Windows Vista, fully-localized versions of Windows were dropped. From Windows Vista onwards, all versions of Windows consist of a base language-neutral version with a language pack installed on top.

While it's true that access to the feature has improved in more recent versions of Windows, the feature has existed for over a decade. But of course, that doesn't stop people from claiming that it's a "new" feature. Don't let the facts get in the way of a good story.

Raymond Chen

Follow

