

Do not write in-process shell extensions in managed code

 devblogs.microsoft.com/oldnewthing/20061218-01

December 18, 2006



Raymond Chen

Jesse Kaplan, one of the CLR program managers, explains [why you shouldn't write in-process shell extensions in managed code](#). The short version is that doing so introduces a CLR version dependency which may conflict with the CLR version expected by the host process. Remember that shell extensions are injected into all processes that use the shell namespace, either explicitly by calling `SHGetDesktopFolder` or implicitly by calling a function like `SHBrowseForFolder`, `ShellExecute`, or even `GetOpenFileName`. Since only one version of the CLR can be loaded per process, it becomes a race to see who gets to load the CLR first and establish the version that the process runs, and everybody else who wanted some other version loses.

Update 2013: Now that version 4 of the .NET Framework supports in-process side-by-side runtimes, is it now okay to write shell extensions in managed code? [The answer is still no.](#)

[Raymond Chen](#)

Follow

