

Drag and drop is a general purpose transfer model, not exclusive to Explorer directories

devblogs.microsoft.com/oldnewthing/20070507-00

May 7, 2007



Raymond Chen

If you've designed your program so that it assumes that the the only thing a user can use drag/drop for is dropping your object into a file system directory, then you've already lost. piers wants to be able to determine the destination of a drag/drop operation. From the description, it appears that what piers really wants is the destination **path**, but who says that the drop destination is a directory? The user might have dropped the item into an email message, onto an FTP site, onto a Web site (via Web Folders), or even a directory on an operating system running inside a virtual machine! The follow-up question makes things even more confusing. If the user drops the files into an FTP site or some other virtual folder, how is your program supposed to be able to restart the transfer? You don't know the user's password on that FTP site. You don't know how to restart that virtual machine and log the user on. And even if you did, you don't know how to write to a directory on a virtual machine; only the virtual machine manager knows how to do that. There are an infinite variety of potential virtual folders out there; I doubt you know how to (or even have the ability to) push your data into each one. Once the user drops the data object, the remainder of the transfer is a private matter between the data source and the drop target. It's not like a data source can tell all drop targets, "I want to take over the transfer," because even if the drop target agreed to it, that still leaves the data source the problem of figuring out how to carry out that take-over. What is my recommendation?‡ The data object in the drag/drop loop should follow the standard shell data object transfer protocol so that users can drop the object into an email message, onto an FTP site, etc.° For the special bonus behavior, I would create a drag/drop hook. A user that wants to do a transfer mediated by your program can use the right mouse button† to drag. When the drop occurs, a context menu will appear, including the drag/drop hook you created. That hook would create a new item called something like "Copy with CoolProgram". (Of course, the hook adds this item only if the data object identifies itself as coming from CoolProgram.) If the user selects "Copy with CoolProgram", then you can do your CoolProgram-mediated copy. **Nitpicker's corner** †More properly, the secondary mouse button, since you may have swapped buttons. ‡Notice that I do not assert that all Microsoft products follow my recommendation. Note also that this is my personal recommendation, not the official position of Microsoft Corporation.

°And you should already understand the standard shell data object transfer protocol before you go off and design a nonstandard one.

Raymond Chen

Follow

