[From pkursawe](http://social.msdn.microsoft.com/Forums/en-US/092c1a9f-74be-4788-860e-5bdede5f2efe/strategies-for-implementing-multiple-activation-contracts-in-an-app?forum=winappswithhtml5)

So as far as I understand it, once my app is running and has been activated once, the same contract will not launch it again. Meaning, as long as the app is not terminated (by the user, system) its state remains intact.

Now, when your app has been "launch" activated, and then is "protocol" activated the state is basically the same for the protocol as the launch version of your app. So I have to protect initializer code to run only once. Otherwise it would run again when the app is activated via "protocol".

ShareTarget activation seems to behave differently, as it does not seem to share the same instance of the app? Am I right?

OK, let’s dive into this.

First of all, what are the contracts that can launch an app? These can be found in the [Windows.ApplicationModel.Activation.ActivationKind](http://msdn.microsoft.com/en-us/library/windows/apps/windows.applicationmodel.activation.activationkind.aspx) enumeration:

* Launch: from the tile
* Search: search contract
* ShareTarget
* File: file association
* Protocol: URI association
* FileOpenPicker (rare—generally for file providers)
* FileSavePicker (rare)
* CachedFileUpdater (rare)
* ContactPicker
* Device (AutoPlay)
* PrintTaskSettings
* CameraSettings
* AppointmentsProvider
* Contact
* LockScreenCall (call apps)
* Phone Only:
  + VoiceCommand
  + PickerReturned
  + WalletAction
  + PickFileContinuation
  + PickSaveContinuation
  + PickFolderContinuation
  + WebAuthenticationBrokerContinuation

Let’s pick the most common subset of these, which would be Launch, Search, ShareTarget, File, Protocol, PrintTaskSettings, and one other for good measure, let’s say ContactPicker (which behaves similarly to other picker scenarios). Then we can test some scenarios and see how they’ll work.

For this exercise I’ll start with a copy of the ContactPicker sample, because it already has that code within it. That is, it has a declaration for Contact Picker that goes to contactPicker.html, whereas default.html is the main start page.

The way you can tell if another instance of the app gets launched is in VS, which will show a second MSAppHost process under Script Documents.

What’s important here is that if you have the *same start page* listed for more than one contract, that you’ll need to clearly separate those code paths. If you get a new instance for the contract, however, then, it’s not a need.

The question that we really want to ask here is—what’s the behavior for any given contract if the same app is and/or isn’t already running?

|  |  |  |
| --- | --- | --- |
| Contract | Not Running | Already Running |
| Launch | Activated, loading Start page in manifest, activated event fired | Switches to the existing instance, no event raised |
| Search | Doesn’t happen in 8.1, because you can only select app scope when invoking Search from that apps. | The app invoked the Search charm and gets events directly. No activation event. |
| ShareTarget | Launched for contract with contract’s start page in Declarations of manifest. | New instance—even if the same Start page is indicated. Generally speaking, you use a self-contained and optimized for a flyout UI. |
| File | Launched for contract with contract’s start page in Declarations of manifest. | Current instance is activated for the contract *if* the current app’s start page (e.g. default.html) is the same as the Start page for the contract. Otherwise you get a new script context. |
| Protocol | Launched for contract with contract’s start page in Declarations of manifest. | Current instance is activated for the contract *if* the current app’s start page (e.g. default.html) is the same as the Start page for the contract. Otherwise you get a new script context. |
| FileOpenPicker FileSavePicker CachedFileUpdater ContactPicker | Launched for contract with contract’s start page in Declarations of manifest. | New instance regardless of Start page, but launches that start page. |
| Device | Launched for default start page in manifest (ignored declaration’s start page) | Activates existing instance *and raises the activated event*. |
| PrintTaskSettings |  |  |
| CameraSettings |  |  |
| AppointmentsProvider | Activated for start page in the manifest | New instance |
| Contact | Same as protocol activation | Same as protocol activation |
| LockScreenCall | Launches | Activates the existing instance—if start page is different: |
| AccountPictureProvider | Goes through protocol activation for “ms-accountpictureprovider” | Same as protocol activations |