HOWTO: Determine if a Scanner is ADF/Flatbed, or Both

With respect to automatic document feeding versus flatbed scanning, there are five "flavors":

- Document Scanners (Automatic Document Feeder only, no flatbed)
- Flatbed Scanners (Scanner with just a flatbed where documents are manually placed one side down at one time)
- Hybrid Document/Flatbed (scanner with a flatbed AND an Auto Document feeder)
- Manual "hand" scanners (scanners where the user physically moves the scan head across a document)
- Cameras with TWAIN interfaces (cameras that allow TWAIN to capture and transfer images as if they were a scanner)

This article is aimed at determining if your scanner is one of the first three (a Manual scanner and a camera will likely act much like a flatbed for this purpose, but we will not be considering them explicitly)

In TWAIN, there's no easy, direct way to ask a scanner "so, do you have flatbed, ADF, or both?"

Instead, we need to determine which of the three types we have by attempting to query, set, and re-query some settings related to the Document feeder. Even then, this is not always a 100% guarantee as every TWAIN device manufacturer and driver writer is free to implement (or not) as much or as little of the TWAIN specification as they wish.... and full adherence "to the letter of the spec" is quite rare.

THE THEORY:

On scanners that have no flatbed, if you attempt to set device.DocumentFeeder.Enabled to false and then check the value of device.DocumentFeeder.Enabled, you'll see it's back to true - that's how you can tell that a scanner is ADF only.

On scanners with no ADF, the value of device.DocumentFeeder.Enabled will be false and if you attempt to set to true, you'll either get an error, or it will just "not take" and when you reinspect after setting it will be false.

On scanners with both ADF and flatbed, you will actually be able to toggle the device.DocumentFeeder.Enabled to true/false and "have it stick"

Well, this is the theory based on how the TWAIN spec is written, but as we said, some scanners don't really honor the spec that well and may let you set the value freely but just not honor it... they could be any of the three types. This is always good to keep in mind when working with TWAIN - that scanners and scanner drivers can vary widely in their support and behavior.

THE CODE:

```
f (device.TryOpen()) { // testing initial state of the DocumentFeeder property if
(device.DocumentFeeder.Enabled) { //we have a doc feeder enabled, try disabling and check
device.DocumentFeeder.Enabled = false; if (device.DocumentFeeder.Enabled) { // setting didn't
stick - this is NOT a flatbed capable scanner // your code to take based on this information
here } else { // you disabled it - chances are there's a flatbed // your code to take based
on this info (that we have both adf and flatbed) // you MAY want to re-enable it here } }
else { // start condition, no device feeder - this may be a flatbed // confirm this by trying
to enable and checking the results device.DocumentFeeder.Enabled = true; if
(device.DocumentFeeder.Enabled) { // setting stuck this scanner DOES have an ADF, and likely
a flatbed too // your code to take based on this information here (that we have both adf and
flatbed) } else { // the setting did not stick, so this means you're flatbed only // your
code to take based on this info (that we have flatbed only) } } // all that code has sorted
out whether you have adf, flatbed, or both... // now, if device.DocumentFeeder.Enabled ==
true, // then you can set other DocumentFeeder properties or general device properties //
you're also connected to the scanner and so can set other properties and perform an Acquire()
if you like.. // remember, your Acquisition object should have handlers for at least //
ImageAcquired and AcquireCancelled and AsynchronousException defined // OPTIONAL
device.Acquire(); // IF you call device.Acquire above then DO NOT call this.. but if you are
not calling device.Acquire, then // Make sure to call device.Close(); when you are done
here.. DO NOT call this AND acquire // Acquire will properly close device when appropriate.
if you cal both acquire and close in order here you wil break things // this close is
"cleaning up" after just determining if flatbed or not device.Close(); } else { // you
couldn't open the scanner, so it may be off or disconnected MessageBox.Show("Could not
communicate with scanner, ensure it's plugged in and turned on and try again"); }
```

You must have the device open for communication in order to make any of these changes/tests. However, we strongly discourage the use of device.Open() because it makes the assumption that the Open() actually worked, and can put some scanner drivers into an odd state if it fails. Instead, use device.TryOpen() that way you can gracefully handle a situation where the scanner driver is present in the scanner list, but isn't connected or isn't powered up/available.

Original Article:

Q10379 - HOWTO: Determine if a Scanner is ADF/Flatbed, or Both

HOWTO: Determine if a Scanner is ADF/Flatbed, or Both

Atalasoft Knowledge Base

https://www.atalasoft.com/kb2/KB/50112/HOWTO-Determine-if-a-Scanner-is-ADFF...