HOWTO: TWAIN Scanning in WPF

DotTwain (Included in DotImage) is a TWAIN scanning component which many customers use to enable TWAIN scanning / acquisition in their windows forms apps.

A significant number of our customers use WPF instead of Windows forms for developing their desktop applications, and so it's common to want to add TWAIN acquisition to a WPF new or existing WPF application.

However, there's a problem. The TWAIN specification itself is directly at odds with the way in which WPF interacts with the Windows Event Loop (Message pump). In practical terms, this means that whether TWAIN acquisition works at all within a given WPF app has a lot to do with how the scanner TWAIN driver was written.

At best, you may find that if you set your

```
device.ThreadingEnabled = false;
```

and

```
device.ModalAcquire = true;
```

you may find that you can coax a given scanner into acquiring successfully in your WPF app. However, this means that scanning needs to take full control of the thread that the UI is running in (Modal acquire) so no other action is going on in the UI / viewer at that time.

So, the bottom line is that WPF and TWAIN do not "play nicely" together by default.

All is not lost ...

WPF Hosting a WinForms User Control

It turns out that WPF allows for Windows Forms Integration (Hosting a WinForms User Control within a WPF app), and using this feature, it is possible to build a Windows Forms User Control / Windows Forms Composite Control which will use DotTwain within an environment that your system TWAIN manager understands, but still allow you to use WPF for the rest of your application.

The basic principles are discussed here:

HOWTO: TWAIN Scanning in WPF

https://msdn.microsoft.com/en-us/library/ms750944%28v=vs.100%29.aspx

There is a sample solution which implements that approach attached to this case as WpfTwainSample.zip

Original Article:

Q10434 - HOWTO: TWAIN Scanning in WPF

Atalasoft Knowledge Base
https://www.atalasoft.com/kb2/KB/50066/HOWTO-TWAIN-Scanning-in-WPF