



# PhotoScan

Written by: Edward Willemsen, 2008

**Table of Contents**

- 1. Introduction..... 3
- 2. Program ..... 3
  - 2.1. Main Menu ..... 7
  - 2.2. Scan Wizard ..... 8
- 3. Settings ..... 8
- 4. Information..... 10
- 5. Techno Talk..... 11
- 6. History ..... 12

## 1. Introduction

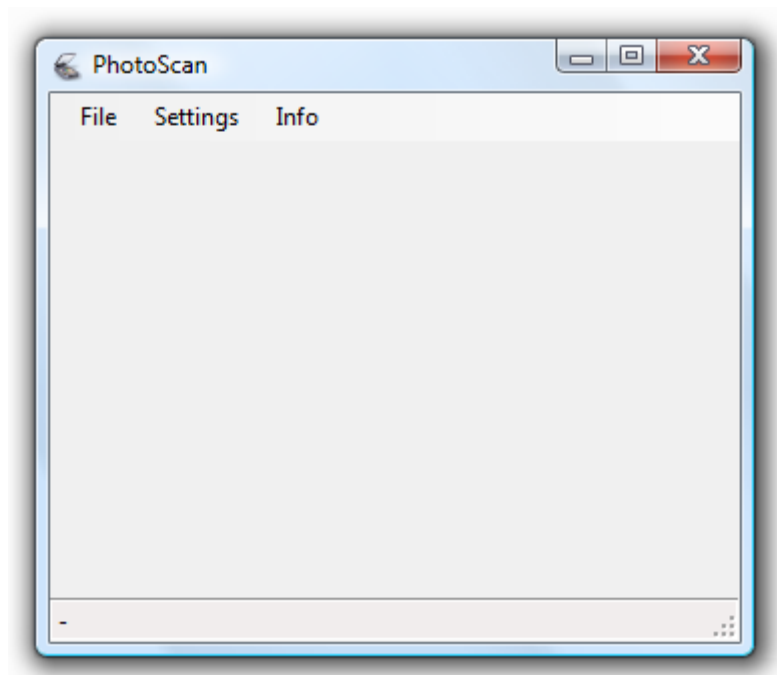
Recently I bought an All-in-One. A regular guy will probably try all features of his newly obtained device, but my first thought was: 'Can I talk with it?'. I'm already familiar with the TWAIN, Technology Without An Interesting Name, concept and was delighted to see that this new device is WIA, Windows Image Acquisition, capable. This application is able to quickly scan a single image from a WIA compatible scanning device.

The application can save the image into the regular formats like Windows Bitmap BMP file, Joint Photographic Experts Group JPEG file, Portable Network Graphics PNG file and a Tagged Image File Format TIFF file.

Since I only have one device I was not able to test the program with other devices. Some of the features made available within the program didn't work on my device. I didn't remove the options because they could work with your device. Still most is functional which makes the program rather useful.

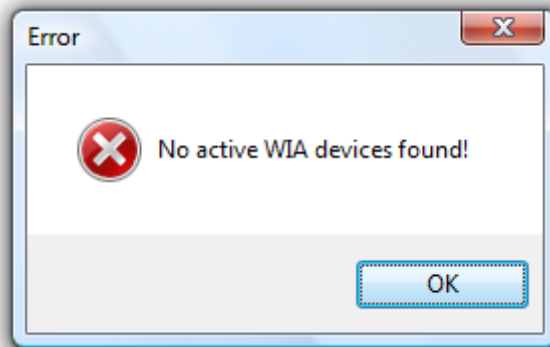
## 2. Program

When you start the application the interface will probably look like this:

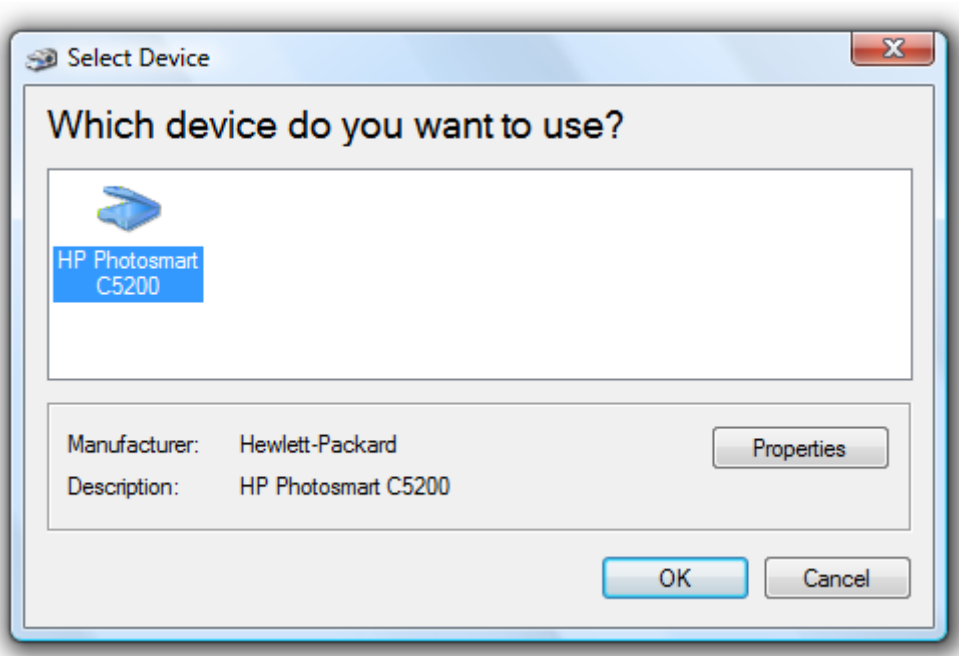


The first thing you'll have to do is to select your device by using the main menu's 'File' -> 'Select Device' option. Be sure that your device is turned on; otherwise it will not be visible.

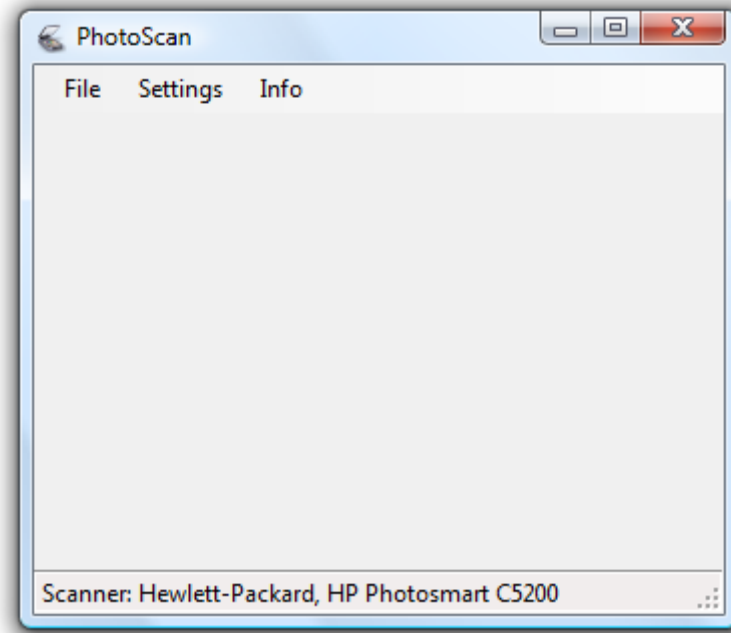
When you just have a single WIA device and it is turned off the following message appears:



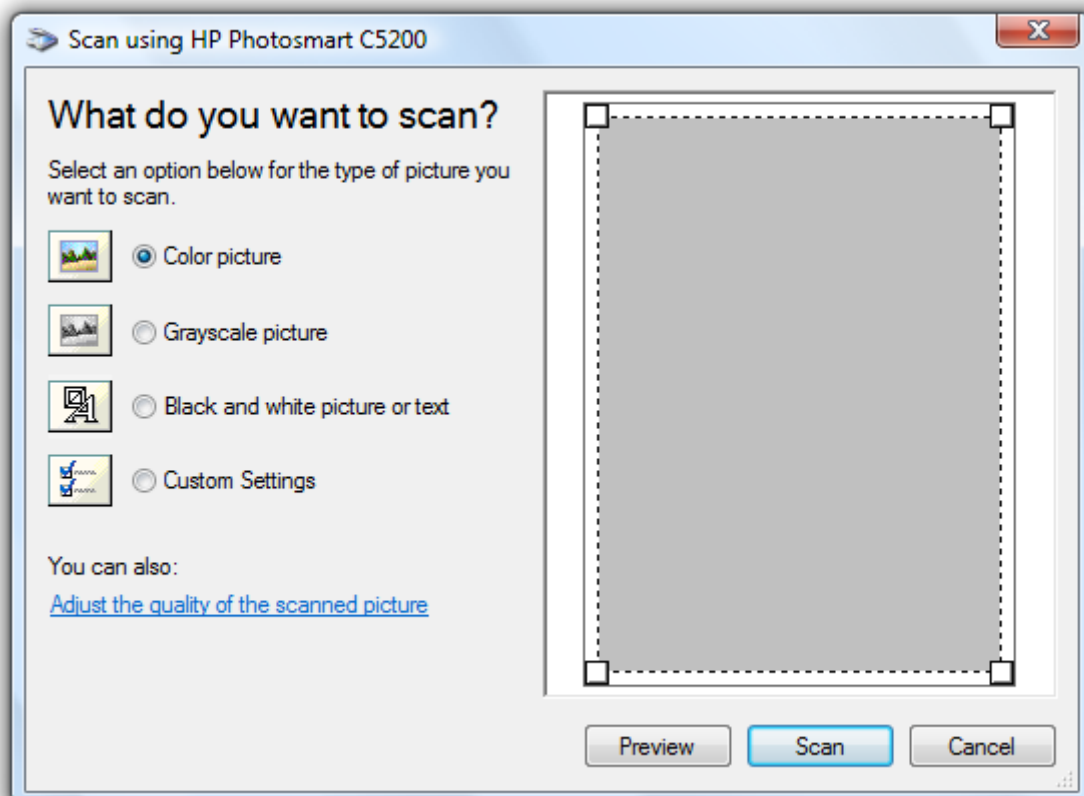
Well, when you device is turned on; the 'Select Device' form is shown and allows you to select the desired device.



When you press the 'OK' button a temporary dialog will appear which shows the communication. The 'Select Device' form will close and the program will show the selected device information in its status strip.



The program is now ready to scan using the 'File'->'Acquire' option found within the main menu, or by pressing the 'Ctrl+A' hotkey. The 'Scan using' dialog appears.

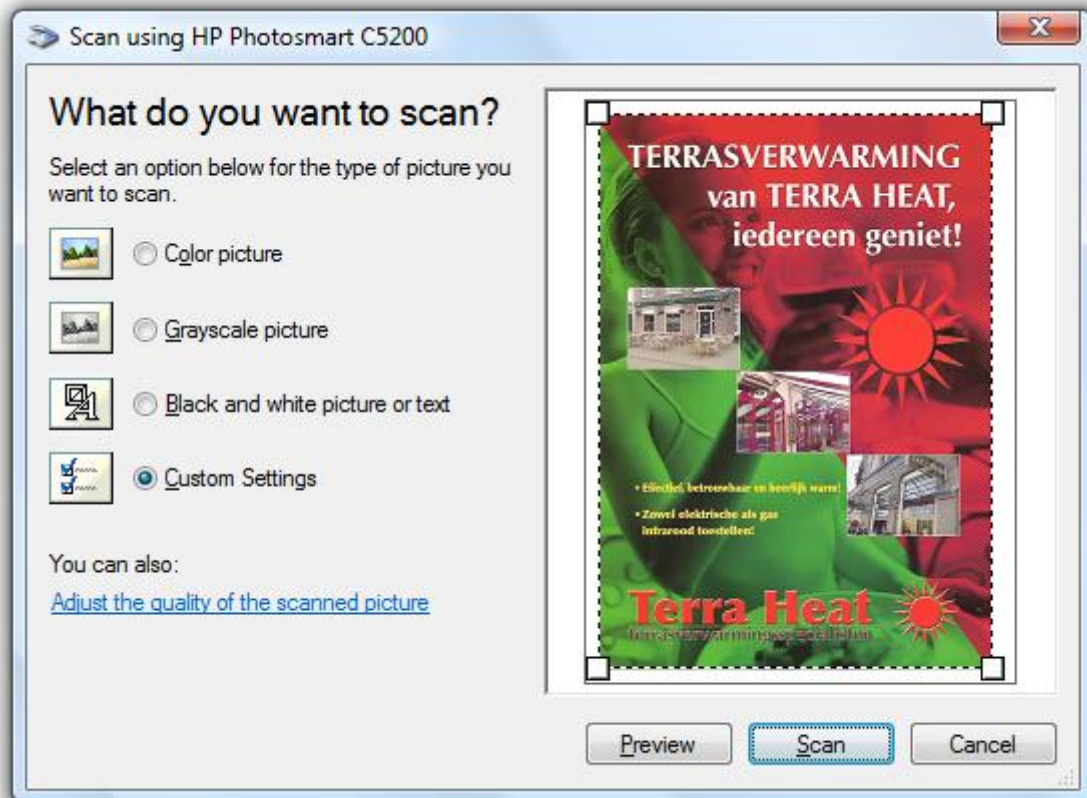


 **Note**

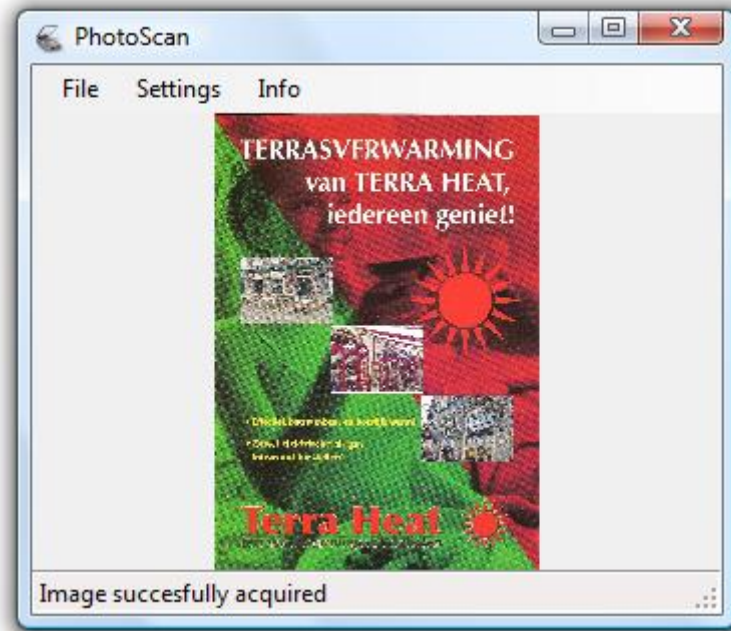
Although you can have several WIA devices which can be selected in the 'Select Device' dialog, the program is only meant for scanner devices and will probably not work well with other types like:

- Cameras (like webcams);
- Video (like TV cards).

By pressing the 'Preview' button within the 'Scan using' dialog, your scanner is going to load a preview of what's lying on the scanner, in our case a flyer of a well known Dutch company.



You can modify the required options and press the 'Scan' button when the image is as required. When the image is scanned a thumb-nailed version will be shown. You can increase the size of this view by increasing the size of the program. The size of the preview has nothing to do with the actual size of the scan.



As you can see, the quality of the thumbnail might look poor, the actual image is not. Next you can use the context menu on the image to use the 'Save as...' option, or use the 'File'->'Save as...' option within the main menu.

The default image type used is JPEG, but can be changed using the settings menu explained later.

The program remembers the last scanned image and you can retrieve it, in its original scanned format by using the 'File'->'Fetch Last Scan' option. This can even be fulfilled without the scanning device turned on.

## 2.1. Main Menu

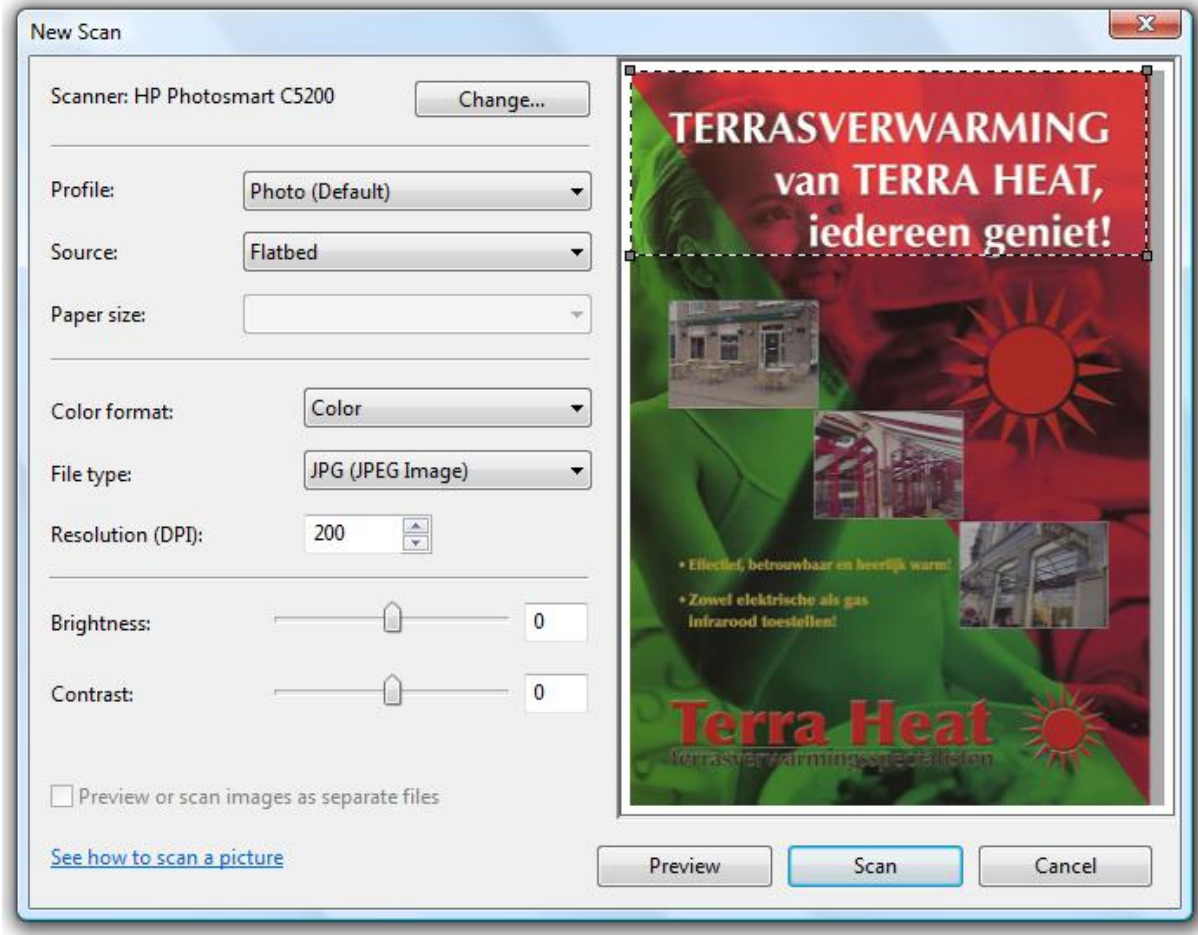
The main menu contains the following items:

- File
  - Acquire
  - Save as...
  - Scan Wizard
  - Select Device
  - Fetch Last Scan
  - Exit
- Settings
  - Encoder Settings
  - Image Codec(s)
  - Scan Preset
- Info
  - Device Properties
  - About

The items will be explained in more detail throughout this document.

## 2.2. Scan Wizard

The scan wizard shows a large form which helps you to scan an image. The result of the scan will not be by the program, but using the built-in wizard. The images will be stored within the Windows Photo 'Recently Imported' picture gallery. For my device the wizard looks like the following:



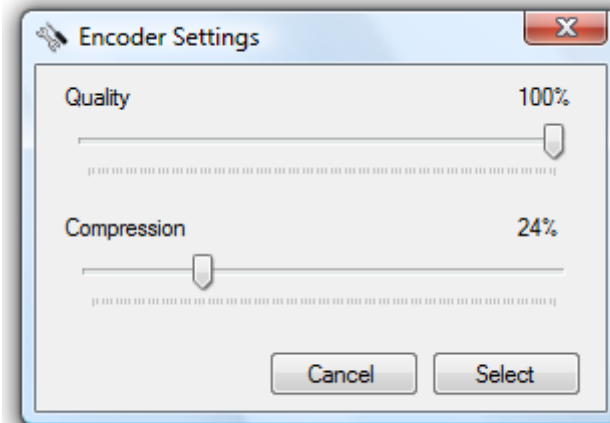
After pressing 'Scan' you'll have to import the image.

## 3. Settings

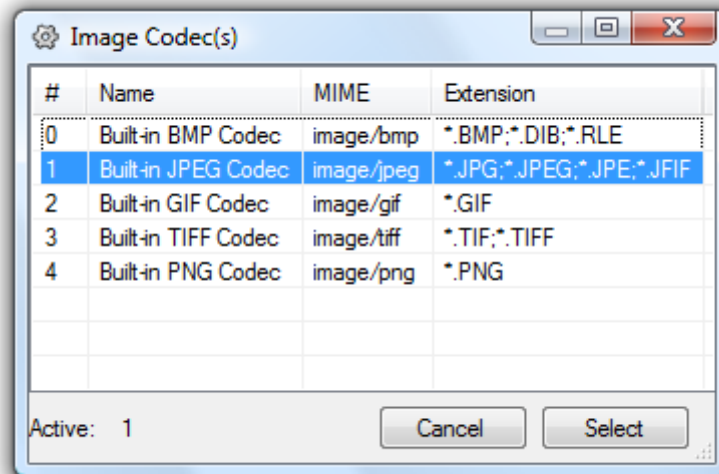
Within the 'Settings' menu found within the main menu, three items can be found, the first one is called 'Encoder Settings' and allows you to change the compression and quality of the saved image. These settings only apply on image formats supporting the features. The second item in the main menu is the 'Image Codec(s)' option which allows you to change the saved image format. And the third one is called 'Scan Preset' and allows you to preset the settings of the scan dialog. As stated in the introduction, not all features work. Well, this dialog is one of them. Although perfectly embedded within the program and call logic, nothing changed while setting the Intent or Quality.



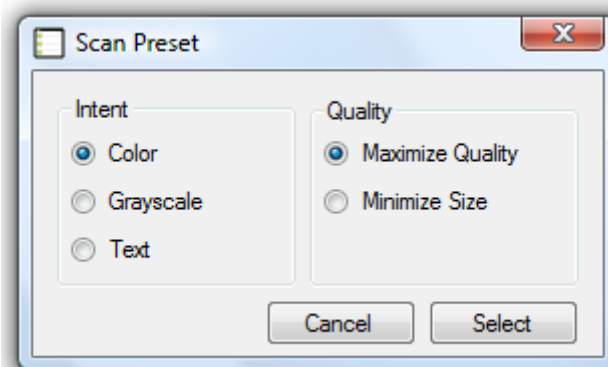
The Encoder Settings form is shown here:



The Image Codec(s) form is shown here:



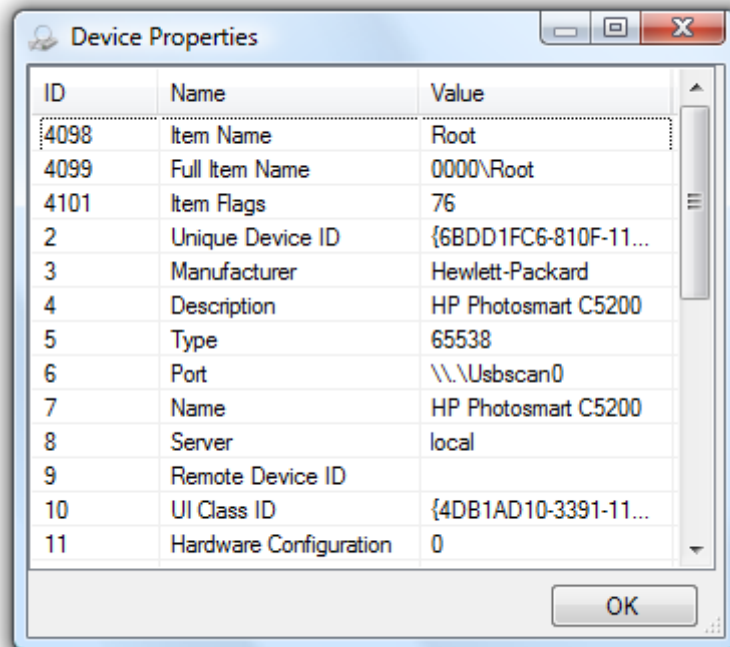
And the Scan Preset form is shown here:



All the settings within these forms will be saved within the application configuration file.

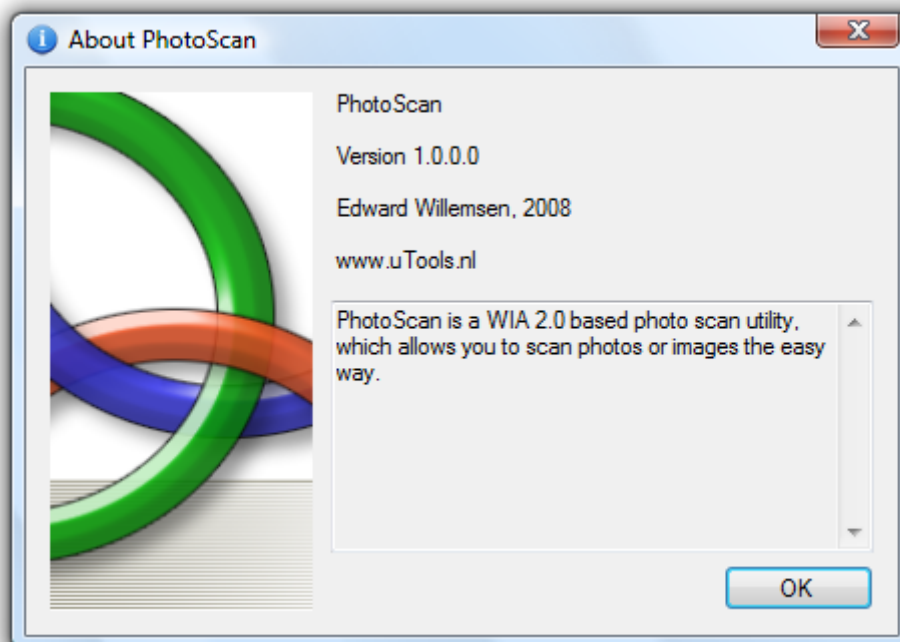
## 4. Information

Within the main menu also the 'Info' option can be found. This option contains two items, with the first one called 'Device Properties'. The 'Device Properties' form shows all the properties of the selected device. For the device used within this document the dialog will show the following properties:



ID	Name	Value
4098	Item Name	Root
4099	Full Item Name	0000\Root
4101	Item Flags	76
2	Unique Device ID	{6BDD1FC6-810F-11...
3	Manufacturer	Hewlett-Packard
4	Description	HP Photosmart C5200
5	Type	65538
6	Port	\\.\Usbscan0
7	Name	HP Photosmart C5200
8	Server	local
9	Remote Device ID	
10	UI Class ID	{4DB1AD10-3391-11...
11	Hardware Configuration	0

The second item found under the 'Info' main menu item is the about box containing the programs version information.



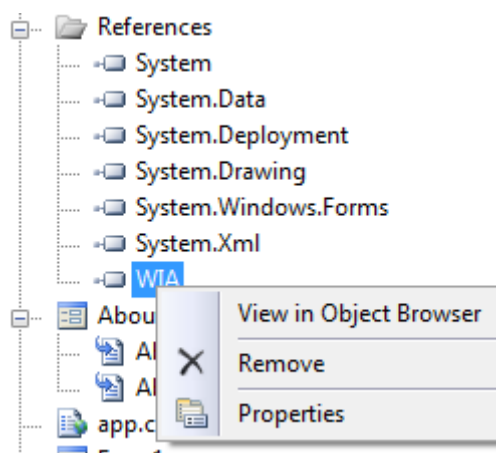
## 5. Techno Talk

It was rather disappointing that not all features seem to work using WIA. I stumbled upon unexpected behavior and features not willing to work. Here is my story.

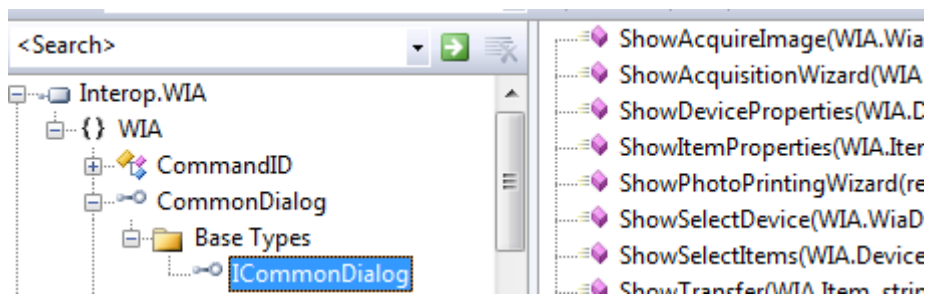
Prior being able to use WIA, you'll have to create a reference toward the WIAAUT.DLL (simply choose 'Project'->'Add Reference' and select the 'Browse'-tab. Go to your SYSTEM32 folder and select the required DLL. Within your references section in your code page add the reference like this:

```
using WIA;
```

One of the cool features of Visual Studio is the fact that you can explore the possibilities within the library. To do this open your Solution Explorer, click References and open the context menu of the WIA reference and select 'View in Object Browser':



Here is a small fragment what can be found within this reference.



As you can see, all the required dialogs callable using WIA can simply be read and used as desired.

When the program starts it will look if the selected device is saved within the user's preferences and it will be attached, sometime like this:

```
DeviceManager manager = new DeviceManagerClass();  
Device d = null;  
foreach (DeviceInfo info in manager.DeviceInfos)  
{  
    if (info.DeviceID == Properties.Settings.Default.DeviceID)  
    {  
        d = info.Connect();  
        foreach (Property prop in d.Properties)  
        {  
            if (prop.Name == "Manufacturer")
```

```

        lblStrip.Text = lblStrip.Text +
            " "+prop.get_Value();
    if (prop.Name == "Description")
        lblStrip.Text = lblStrip.Text +
            ", "+prop.get_Value();
    }
}
}

```

Strange enough, the properties are not available as they should:

```
d.Properties["Description"].Value;
```

I found several Visual Basic snippets using the VB slang for this, but it simply doesn't work in C#.

The next 'amazing discovery' is the following routine:

```

wiaClass.ShowAcquireImage(d.Type,
    PhotoScan.Properties.Settings.Default.PreIntent,
    PhotoScan.Properties.Settings.Default.PreQuality,
    FormatID.wiaFormatBMP, false, true, true);

```

I expected that the FormatID will change image format, but all what is delivered is Windows Bitmap. If you change the FormatID, the file extension will change, but the binary format remains the same. Next issue with this method is the fact that you can feed it with any WialmageIntent or WialmageBias, the dialog content will not change. Well, that's what I experienced with my device.

So how do I save to other image formats? Simple, by using the built-in encoders which will like something like this:

```

Bitmap bmp = new Bitmap(scan.Image);
EncoderParameters enc = new EncoderParameters(2);
EncoderParameter quality = new EncoderParameter(Encoder.Quality,
    Settings.Default.Quality);
EncoderParameter compression = new
EncoderParameter(Encoder.Compression, (int)
Settings.Default.Compression);

enc.Param[0] = quality;
enc.Param[1] = compression;

bmp.Save(dlgSave.FileName, codec[Settings.Default.Codec], enc);

```

Most of the used encoder parameters will only work on the JPEG image format.

## 6. History

Version	Comment
1.0	The first edition of PhotoScan.

Find more useful utilities and C# code here: <http://www.utools.nl/>