Extracting GPS coordinates from a photo and plotting it on a map

devblogs.microsoft.com/oldnewthing/20131125-00

November 25, 2013

Raymond Chen

Today's Little Program extracts GPS coordinates from a photo and plots it on a map. Remember, Little Programs do little to no error checking, because that's how they roll.

```
#define STRICT
#define UNICODE
#define _UNICODE
#include <windows.h>
#include <shlobj.h>
#include <shellapi.h>
#include <propidl.h>
#include <propkey.h>
#include <propvarutil.h>
#include <atlbase.h>
#include <atlalloc.h>
#include <strsafe.h>
void OpenMap(double dblLatitude, double dblLongitude)
{
 wchar_t szUrl[1024];
 StringCchPrintf(szUrl, ARRAYSIZE(szUrl),
  L"http://www.bing.com/maps/default.aspx?v=2&q=%f,%f",
  dblLatitude, dblLongitude);
 ShellExecute(nullptr, nullptr, szUrl, nullptr, nullptr, SW_NORMAL);
}
```

We start with a simple function that takes a latitude and longitude and opens a Web page that highlights that coordinate. In a real program, you probably would do something more interesting with the coordinates, but I'm opening a Web page just to do *something*.

```
class CPropVariant : public PROPVARIANT {
public:
 CPropVariant() { PropVariantInit(this); }
 ~CPropVariant() { PropVariantClear(this); }
};
```

The `CPropVariant` class is an incredibly lame wrapper around `PROPVARIANT` for RAII purposes.

```
HRESULT GetGPSCoordinateAsDecimal(
    IShellItem2 *psi2,
    REFPROPERTYKEY pkey,
    REFPROPERTYKEY pkeyRef,
    double *pdbl)
{
 CPropVariant spvar;
 HRESULT hr = psi2->GetProperty(pkey, &spvar);
 if (FAILED(hr)) return hr;
 double rgdbl[3];
 ULONG cElt;
 hr = PropVariantToDoubleVector(spvar, rgdbl, 3, &cElt);
 if (FAILED(hr)) return hr;
 if (cElt != 3) return E_INVALIDARG;
 double coord = rgdbl[0] + rgdbl[1] / 60.0 + rgdbl[2] / 60.0 / 60.0;
 CComHeapPtr<wchar_t> spszDir;
 hr = psi2->GetString(pkeyRef, &spszDir);
 if (FAILED(hr)) return hr;
 if (spszDir[0] == L'W' || spszDir[0] == L'S') coord = -coord;
 *pdbl = coord;
 return S_OK;
}
```

The `GetGPSCoordinateAsDecimal` function is where the real work happens. GPS latitude and longitude are encoded in the shell property system as a bunch of related properties.

Property	Type	Meaning
System.GPS.DestLatitudeNumerator	UINT[3]	numerators for degrees, minutes, and seconds of latitude
System.GPS.DestLatitudeDenominator	UINT[3]	denominators for degrees, minutes, and seconds of latitude
System.GPS.DestLatitude	double[3]	degrees, minutes, and seconds of latitude (numerator ÷ denominator)
System.GPS.DestLatitudeRef	string	“N” or “S”
System.GPS.DestLongitudeNumerator	UINT[3]	numerators for degrees, minutes, and seconds of longitude
System.GPS.DestLongitudeDenominator	UINT[3]	denominators for degrees, minutes, and seconds of longitude
System.GPS.DestLongitude	double[3]	degrees, minutes, and seconds of Longitude (numerator ÷ denominator)
System.GPS.DestLongitudeRef	string	“E” or “W”

Each of the coordinates is recorded in DMS form as pairs of unsigned integers (numerator and denominator). The direction is recorded as a string as a separate property. Why this wacky format? Probably because that's the way EXIF records it.

For convenience, there is a combo property which does the division for you (but frustratingly, does not flip the sign for direction). And if you want the coordinates in decimal form, then you'll have to do the DMS-to-decimal conversion yourself.

We start by getting the DMS value as a `PROPVARIANT` then converting it to an array of `double`s. (There had better be three of them.) We then use the power of mathematics to convert from DMS to decimal degrees.

Finally, we flip the sign if the direction from center is West or South.

Now it's time to put these functions together.

```
int __cdecl wmain(int argc, wchar_t **argv)
{
 if (argc < 2) return 0;
 CCoInitialize init;
 CComPtr<IShellItem2> spsi2;
 if (FAILED(SHCreateItemFromParsingName(argv[1],
             nullptr, IID_PPV_ARGS(&spsi2)))) return 0;
 double dblLong, dblLat;
 if (FAILED(GetGPSCoordinateAsDecimal(spsi2, PKEY_GPS_Longitude,
                   PKEY_GPS_LongitudeRef, &dblLong))) return 0;
 if (FAILED(GetGPSCoordinateAsDecimal(spsi2, PKEY_GPS_Latitude,
                   PKEY_GPS_LatitudeRef, &dblLat))) return 0;
 OpenMap(dblLong, dblLat);
 return 0;
}
```

Find a photo with GPS information encoded inside it and pass it on the command line as a fully-qualified path. (Because I'm too lazy to call `GetFullPathName`.) The program should open a Web page that shows where the picture was taken.

Raymond Chen

Follow