Connecting some blocks: Tell me more about the current image in my wallpaper slide show

devblogs.microsoft.com/oldnewthing/20150302-00

March 2, 2015

Raymond Chen

One of my colleagues said,

> I really like the wallpaper slide show feature, especially the one that rotates through the top-rated pictures in my photo library. My photo library includes photos taken by other members of my family, and sometimes I'll get a wallpaper that I want to learn more about. It'd be great if there were some way to say "Hey, tell me more about this wallpaper."

Today's Little Program snaps together some blocks.

- It registers a command on the desktop background.
- The command looks at the current wallpaper image.
- And then shows you some information about it in the form of a property sheet.

(My colleague actually preferred that the image be opened in Photo Gallery, but I'll just show the properties because not every has Photo Gallery installed, and the purpose of the exercise is to show how to snap blocks together, not to argue about which color blocks to use.)

```
#define UNICODE
#define _UNICODE
#define STRICT
#include <windows.h>
#include <shlobj.h>
#include <shellapi.h>
#include <atlbase.h>
#include <atlalloc.h>

void ShowProperties(PCWSTR pszFile)
{
 SHELLEXECUTEINFO sei = { sizeof(sei) };
 sei.fMask = SEE_MASK_FLAG_DDEWAIT | SEE_MASK_INVOKEIDLIST;
 sei.nShow = SW_SHOWNORMAL;
 sei.lpVerb = L"properties";
 sei.lpFile = pszFile;
 ShellExecuteEx(&sei);
}

int WINAPI wWinMain(HINSTANCE, HINSTANCE, LPWSTR, int)
{
 CCoInitialize init;
 ProcessReference ref;

 // The rest of this code is adapted from
 // Obtaining information about the user's wallpaper on multiple monitors

 CComPtr<IDesktopWallpaper> spdw;
 CoCreateInstance(CLSID_DesktopWallpaper, nullptr, CLSCTX_ALL,
                  IID_PPV_ARGS(&spdw));

 CComHeapPtr<wchar_t> spszCommonWallpaper;
 HRESULT hr = spdw->GetWallpaper(nullptr, &spszCommonWallpaper);
 if (hr == S_OK) {
  // Single wallpaper on all monitors.
  ShowProperties(spszCommonWallpaper);
 } else if (hr == S_FALSE) {
  // Different wallpaper on each monitor.
  // Look for the monitor that has the mouse pointer.
  POINT pt;
  GetCursorPos(&pt);

  UINT count;
  spdw->GetMonitorDevicePathCount(&count);

  for (UINT i = 0; i < count; i++) {
   // Get the device path for the monitor.
   CComHeapPtr<wchar_t> spszId;
   spdw->GetMonitorDevicePathAt(i, &spszId);

   // Get the monitor location.
   RECT rc;
   spdw->GetMonitorRECT(spszId, &rc);
   if (PtInRect(&rc, pt)) {
    // Get the wallpaper on that monitor.
    CComHeapPtr<wchar_t> spszWallpaper;
    hr = spdw->GetWallpaper(spszId, &spszWallpaper);
    ShowProperties(spszWallpaper);
    break;
   }
  }
 }

 return 0;
}
```

Now we get to hook this up to the context menu for the desktop.

```
[HKEY_CLASSES_ROOT\DesktopBackground\shell\Wallpaper properties\command]
@="WallpaperProperties.exe"
```

Now you can right-click on the desktop background and select *Wallpaper properties*, and it will show you the properties of the wallpaper. You are of course welcome to do something else, like load the wallpaper into your favorite image viewer.

Note that I cheated a bunch here. In addition to pretty much ignoring all error checking (because this is a Little Program), I also use the current mouse position instead of using the point at which the context menu was invoked.

Raymond Chen

Follow