Subject: Register new file types under windows
Posted by ptDev on Sat, 20 Jun 2009 09:50:44 GMT
View Forum Message <> Reply to Message

Hi to all,

I just figured out together the Windows Registry calls needed, and made a couple of functions that make it easy to create a new "document type" for Windows Explorer to recognize, and tell the shell which program to launch. All you need to do with these is pass Strings as arguments, and the Windows Registry will be updated accordingly.

The first function has two required arguments (the file extension, in the ".ext" format, and the full program executable path, e.g. "C:\\Program Files\\My Program\\prog.exe"). The rest is optional (defaults to ""): a "Document Class" you might wish to register (a common name for this is "MyProgram.ext"); a description String that Windows Explorer uses to describe the document type; and a full path to an icon resource (it usually is an icon resource that is embedded in your executable, e.g. "prog.exe,1"). Note that if this last parameter is not supplied, Windows automatically renders an icon, by composing the "generic file" with your executable's icon.

```
bool WinRegisterDocType(const String &extension, const String &program,
 const String &doc_class = "", const String &description = "", const String &icon_path = "")
{
 const String command = "\"" + program + "\" \"%1\"";
 String pathbase;

 if(!SetWinRegString(doc_class, "", extension, HKEY_CLASSES_ROOT))
  return false;

 if(doc_class.IsEmpty())
  pathbase = extension;
 else
  pathbase = doc_class;

 if(!SetWinRegString("open", "", pathbase + "\\shell", HKEY_CLASSES_ROOT))
  return false;
 if(!SetWinRegString(command, "", pathbase + "\\shell\\open\\command",
HKEY_CLASSES_ROOT))
  return false;

 if(!description.IsEmpty())
  SetWinRegString(description, "", pathbase, HKEY_CLASSES_ROOT);
 if(!icon_path.IsEmpty())
  SetWinRegString(icon_path, "", pathbase + "\\DefaultIcon", HKEY_CLASSES_ROOT);

 return true;
}
```

The second function is meant to reverse the above process and un-install a document type. It only

needs the extension string (same convention as above), and an optional document class name string.

```
void WinUnregisterDocType(const String &extension, const String &docclass = "")
{
 DeleteWinReg(extension, HKEY_CLASSES_ROOT);
 if(!docclass.IsEmpty())
  DeleteWinReg(docclass, HKEY_CLASSES_ROOT);
}
```

ATTENTION: These functions require administrator priviledges to work.

I hope someone else finds these useful.

EDIT: Simplified the code substantially.

Regards,
Francisco
