ShellExtDatedFolder

A Delphi Windows Explorer Context Menu Extension to Create Dated Folders

Genesis

Talking with a colleague he expressed a need for a way to easily create new folders that are named using today's date. I responded that it should be a piece of cake. I was very wrong. While it is possible to create new folders in a Windows Shell with but a single entry or two in the System Registry, you can't control the date format easily because of differences in localization. You really need some code to achieve your ends.

That started me on the path to learning about Windows Context Menu Extensions, that are implemented using COM. That, of course, meant I had to learn something about COM. The whole process was lengthy, frustrating, but also rewarding. The results of my explorations can be found in the associated GitHub project (see link below.)

Obtaining the Source Code

The project source code can be downloaded as a .zip file or cloned as a Git project from `ShellExtDatedFolder`. (GitHub)

Objectives

Here's what I hoped to accomplish. Most of this is a learning path, something I really enjoy doing.

1. Learn how to develop COM Servers (which is what MS Shell Extensions are)
2. Learn the specific requirements for various Explorer plugins
3. Study and dissect the Marco Cantu and Embarcadero examples
4. Learn a little about the C++ implementation of this
5. Develop a `Windows Explorer` Context Menu Extension that creates a new folder named using the current system date. This is a convenience and accuracy improvement for the user who needs to repeatedly create such folders as a means of organizing quantities of individual files by date.

First Steps

1. Establish repo.
2. Move existing Cantu and Embarcadro examples into the Repo.
3. Restructure the project libraries to facilitate easy study and analysis.
4. Perform code analysis. Code analysis consists of two major activities:
    - Read existing code and research all unknown APIs, structures and related Microsoft documentation.
    - Comment the existing code. Commenting code strengthens understanding and cements what was learned from research. By commenting, I mean, almost every line and in the case of complex operations, paragraphs of comments associated with the methods, classes or other parts of the code.
5. Using what was learned during the code analysis, implement a Context Menu that creates a folder that includes a name with the current date.

The ShellExtDatedFolder Project Group

- All projects in the project group are Delphi projects, produced with RAD Studio Enterprise 10.4.
- All code is heavily commented. This enables the code to be used as a learning tool.
- All code is contained in this project group. There is one project in the project group:

ZNewFolder Project

Downloading the project code will permit examination of the actual code along with the code comments. The code comments will clarify much of what may be ambiguous here.

Overview

The project builds a COM .dll that populates a Shell Context Menu Extension with a single additional menu item, `Zilch Dated Folder`. The menu item is only displayed when a right mouse click is made in a folder background or folder tree folder. When the user selects the item, a new folder is created within the selected folder named `Zilch-yyyy-mm-dd` where `yyyy-mm-dd` is replaced by the current system date.

Implementation

A standard, lightweight COM object was created using the appropriate Delphi Wizard. The `TZNewFolderExt` CoClass was created by the Wizard. Microsoft Shell Extensions for Context Menus require the implementation of two Interfaces.

IShellExtInit (Interface)

`IShellExtInit` has a single method, `Initialize`. `Initialize` is called by the Shell after the CoClass is instantiated. It passes three values that can be utilized by the CoClass during later methods:

pidlFolder

is a `PItemIDList` that contains information about the folder container that was right-clicked.

lpdobj

is a pointer to the interface `IDataObject` that contains a list of all objects in the folder that was right-clicked.

hkeyProgID

is a Registry Key Handle `HKEY` to the program.

Of these, only `pidlFolder` is of current interest. From it is extracted the full file path to the folder that will receive the newly created folder. This information is saved as a string in a CoClass property.

IContextMenu (Interface)

`IContextMenu` has three methods:

`QueryContextMenu`

is used to specify the desired menu item position and text.

`GetCommandString`

is optional. It is used to specify status bar help text and canonical names of verbs in Windows XP and earlier versions of Windows.

`InvokeCommand`

is invoked when the user actually chooses (clicks) the menu item it supports.

Registration

An additional class, `TZNewFolderExtFactory`, was also added that descends from `TComObjectFactory`. This permits the extension of the `UpdateRegistry` procedure to include three additional System Registry keys to support Context Menu Extensions.

Installation

In all cases, installation may require elevated authorization. Running as `Administrator`, either when using the `cmd` window or the installation file should be sufficient to achieve correct installation.

Installation can be manually achieved by copying the ZNewFolder.dll to an appropriate folder and then using RegSvr32.exe (a Microsoft program) to register the COM server. The registration process handles all necessary System Registry modifications. RegSvr32 can also be used to uninstall the Com server using the -u switch.

Also included is an installation file created using `DeployMaster`, an inexpensive installation package builder available from the `DeployMaster` website. `Deploymaster` also provides an uninstall function.

References

I found a number of useful references while working on this. Here are some I found useful.

Component Object Model (COM)

Microsoft Documentation
The Component Object Model documentation portal page. If you have time and are trying to learn about COM a quick read of this documentation (all pages) might be what you need. It's long, and assumes knowledge of C++, so it's not easy reading.

Developing COM-Based Applications Index

Embarcadero DocWiki
The COM development portal page. Provides information specific to Delphi. This is also long but since it has information devoted to Delphi it may prove easier to digest.

The Complete Idiot's Guide to Writing Shell Extensions - Index

Idiot's Guide
An index of all the articles in the Idiot's Guide Since the Idiot's Guide series has become quite large, here is an index of all the articles that gives a quick blurb about each one, so you can quickly find the articles that interest you.
Unfortunately, this suffers from two problems:

- The content is somewhat dated, although it still contains much valuable information.
- It is not Delphi-centric. It is written for C++ programmers so if your strictly interested in Delphi this is probably not your cup of tea.

Creating Shortcut Menu Handlers

Microsoft Documentation
A significant and lengthy discussion of Shortcut Menu Handlers. Difficult to understand without a lot of background understanding, but very important to do so.
Shortcut menu handlers, also known as context menu handlers or verb handlers, are a type of file type handler. Like all such handlers, they are in-process Component Object Model (COM) objects implemented as DLLs.

Writing a Windows Shell Extension

Marco Cantù
Delphi Product Manager Marco Cantù's blog post. Also links to a video. This focuses on file operations and COM rather than Menu Shortcuts. Still contains lots of general information useful to understand Shell development.

Important Caveats

Bitness

If you are running a 32-bit Shell, you must register and use the 32-bit version of the Com server. On a 64-bit system you must register and use the 64-bit version.

The `DeployMaster` installation program contains both versions and will correctly choose and install the correct version after examining the bitness of the installation system.

Target Systems

The existing code has only been tested on a 64-bit Windows 10 Pro system, version 2009, released October 2020 and code-named 20H2. The installation program will also allow installation on Windows 7 and windows 8 machines, both 32-bit and 64-bit, as well as a variety of Windows servers, but since the code has never been tested on those configurations it is not known if it will function correctly.

Licensing