ShellExtDatedFolder

A Windows Explorer Extension to Create Dated Folders From the Context Menu

Licensing

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

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

First Steps

1. Establish repo.
2. Move existing Cantu and Embarcadro examples into the Repo.
3. Restructure the project libraries to something sensible and easy to maintain.
    - Put individual projects in their own directories.
    - If the project group has mixed language (e.g., Delphi and C++) remove the main .gitignore file from the project group and put separate .gitignore files into each individual project folder, tailored to its contents.
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 code is contained in this project group. There is one project in the project group:

ZNewFolder Project

Overview

The project builds a COM .dll that populates a Shell Context Menu Extension with a single additional menu item, `Zilch Dated Folder`. When the user selects the item, that is only displayed when a right mouse click is made in a folder background or folder tree folder, 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 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 addtional System Registry keys to support Context Menu Extensions.

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.

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.