



A WPF File ListView and ComboBox

Thomas Willwacher

28 Mar 2011 CPOL

The article describes a simple folder navigation control.

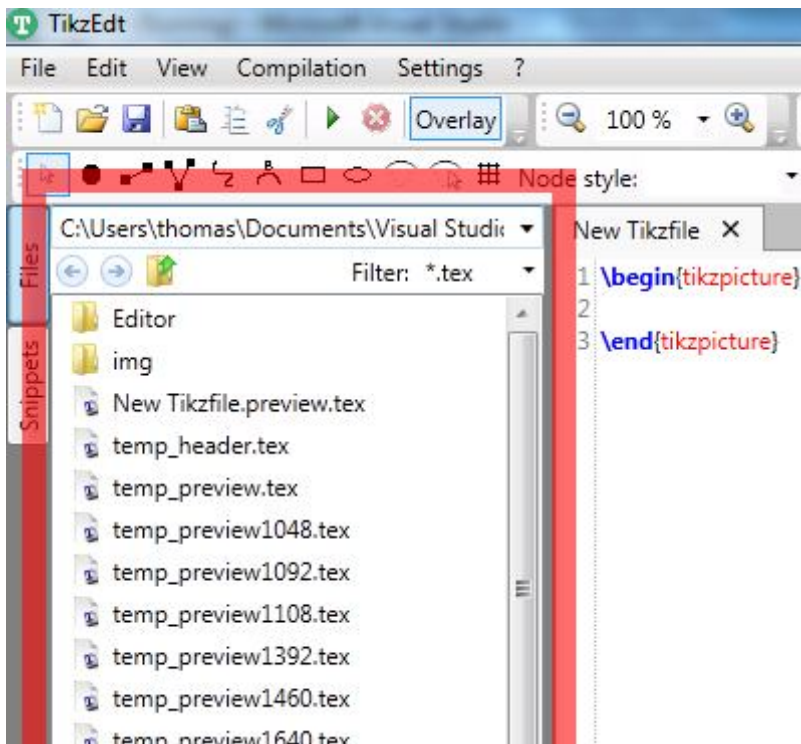
[Download source code - 47.6 KB](#)

Introduction

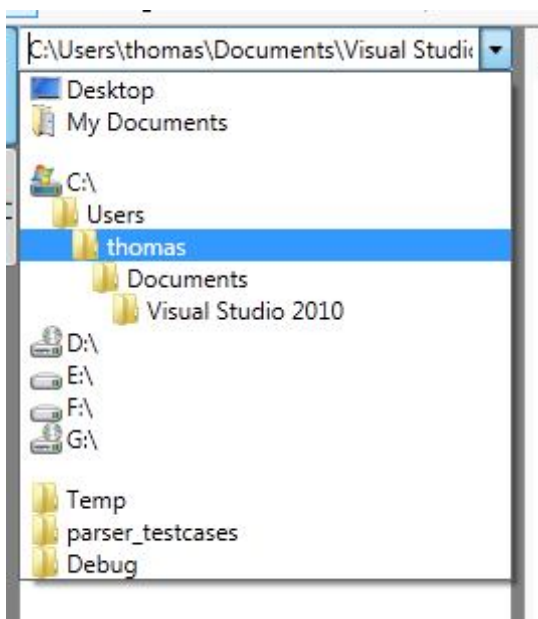
There are many file listviews or file system navigation controls out there. Most are written to resemble the Windows Explorer. I needed a small and simple one for use in an application. It should be displayed in a narrow column on the left of the workspace and allow the user to open files in the current working directory, or close to it. The requirements are:

- The control should use very little horizontal space, see the screenshot.
- The control should be as lightweight as possible, and not clutter the user interface too much.
- It should bind to a most recently used (MRU) file list, and allow the user to jump to locations of those files.
- It should **not** be an Explorer clone, and the user should not be allowed to do Shell operations like copying, renaming files, etc. If you are looking for an Explorer clone, you can find an excellent one [here](#).

Here is how the result looks, embedded in a larger UI, from which only a part is shown:



Here is a closer look at the combobox dropdown:



The three folders close to the bottom are extracted from the application's MRU list. The full paths of these items are not displayed, but are shown in a tooltip when hovering the mouse over.

Using the code

Download the above .zip-file and add a reference to the .dll to your project. The following snippet illustrates how to use the control:

XML

```
<Window ...
    xmlns:FLV="clr-namespace:FileListView;assembly=FileListView"
    ... >
```

```

...
<FLV:FolderView Filter="*.tex|*.txt|*.exe|*.*"
    RecentLocations="{Binding RecentFiles}"
    OnFileOpen="folderView_OnFileOpen"/>
...

```

Here, **RecentFiles** is your MRU list. The control extracts the directories from this list and removes the duplicates automatically. When the user selects a file by double click or pressing Return, the event **OnFileOpen** is fired. The following additional properties are available:

CurrentFolder The currently displayed folder.
FilterIndex The currently selected filter.
ShowFolders Turns on/off the display of folders.
ShowHidden Turns on/off the display of hidden files.
ShowIcons Turns on/off the display of file and folder icons.

Implementation details

Implementing a file listview is more or less a standard task, and there are only a few things about the code I should comment on. One is that I tried to keep everything lean and stuck with the onboard C# tools where possible. (The final library is only 31 KB small.) One exception is the use of Shell routines to extract associated file and folder icons. Another trick which speeds up display of larger folders a lot is to lazily query the icons associated to files, like this:

C#

```

private ImageSource _DisplayIcon = null;
public ImageSource DisplayIcon
{
    get
    {
        if (_DisplayIcon == null)
        {
            try
            {
                if (type == FSItemType.Folder)
                    _DisplayIcon =
                        IconExtractor.GetFolderIcon(FullPath).ToImageSource();
                else
                    _DisplayIcon = IconExtractor.GetFileIcon(FullPath).ToImageSource();
            }
            catch (Exception e)
            {
            }
        }

        return _DisplayIcon;
    }

    set { _DisplayIcon = value; }
}

```

Similarly, the combobox dropdown list is only constructed when the user actually drops it down.

Links

In a previous version, I used (a slightly modified version of) [this](#) code. A very nice collection of Shell controls and routines has been released by Microsoft [here](#).

History

- 03/12/2011: Rewrote the icon extraction routines in order to also get the associated folder icons. Fixed an issue with the listview column width.
- 03/28/2011: Implemented the listview as a separate control. Used the `AutoGrayImage` class from [here](#).

License

This article, along with any associated source code and files, is licensed under [The Code Project Open License \(CPOL\)](#)

Written By

Thomas Willwacher

 United States

This member has not yet provided a Biography. Assume it's interesting and varied, and probably something to do with programming.

Comments and Discussions

 **12 messages** have been posted for this article Visit

<https://www.codeproject.com/Articles/167873/A-WPF-File-ListView-and-ComboBox> to post and view comments on this article, or click [here](#) to get a print view with messages.

[Permalink](#)

[Advertise](#)

[Privacy](#)

[Cookies](#)

[Terms of Use](#)

Article Copyright 2011 by Thomas
Willwacher

Everything else Copyright ©
[CodeProject](#), 1999-2022

Web01 2.8:2022-09-02:1