File manipulation with IsolatedStorageFile on Windows Phone

This code example demonstrates how to store files locally using Windows Phone Isolated Storage. Official documentation on the topic(s) is available here: Local Data Storage for Windows Phone (MSDN).

Tip: While `IsolatedStorageFile` works in Windows Phone 7 and 8, if you're focussing on Windows Phone 8 only you should use the new API - see Quickstart: Working with files and folders in Windows Phone 8

Introduction

Windows Phone 7 apps store their local data in an app-specific file system area known as Isolated Storage. There are three ways we can store data:

- IsolatedStorageFile - stores files
- IsolatedStorageSettings - stores settings as name/value pairs in a dictionary
- LINQ to SQL - store relational data in a local database.

Note: Symbian developers will recognise that Isolated Storage is conceptually similar to an app's private data cage. The main difference is that Windows Phone apps only have access to their own isolated storage - there is no *shared file system*. This creates challenges for apps that want to share data or files.

This code examples uses IsolatedStorageFileStream class to write and retrieve text (provided by the user) to a file. In a following section, we also provide information about Isolated Storage Explorer, a tool for exploring the file structure in isolated storage.

File Manipulation in WP7

Implementation

When we are ready with an empty project for Windows Phone, let's take some data from user input and save it to a file in Isolated Storage.

(C) Copyright Nokia 2014. All rights reserved.

```
IsolatedStorageFile fileStorage = IsolatedStorageFile.GetUserStoreForApplication();
StreamWriter Writer = new StreamWriter(new IsolatedStorageFileStream("TestFile.txt",
FileMode.OpenOrCreate, fileStorage));
Writer.WriteLine(textBox1.Text);
Writer.Close();
```

We have created an object of IsolatedStorageFile which obtain a virtual storage for the application. Then let's create a new StreamWriter, to write the file to the specified location. `WriteLine()` is used to write the content to the file. And Finally we close the StreamWriter. This will just create a new file with the name **TestFile.txt** in the Isolated Storage.

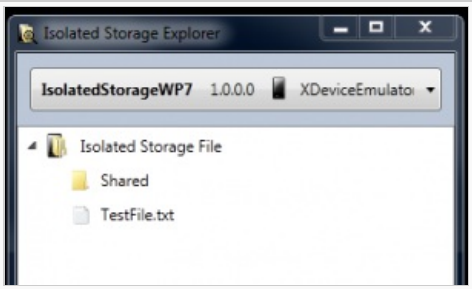
To retrieve the content of the file from the Isolated Storage. Let's again create a virtual store for the application. This time we will create a StreamReader to read the content.

```
IsolatedStorageFile fileStorage = IsolatedStorageFile.GetUserStoreForApplication();
StreamReader Reader = null;
try
    {
    Reader = new StreamReader(new IsolatedStorageFileStream("TestFile.txt",
FileMode.Open, fileStorage));
    string textFile = Reader.ReadToEnd();
    textBox2.Text = textFile;
    Reader.Close();
    }
catch
    {
    MessageBox.Show("File it not created");
    }
```

Then read the file from the specific location. Read the content of the file with `ReadToEnd()` and finally display the content of the file to the Textbox. If the file doesn't exist and user clicks on the retrieve button to display the file content then we display a message. We can also check the file existence with `IsolatedStorageFile::FileExists(FilePath)`.

Windows Phone 7 Isolated Storage Explorer

Windows Phone 7 Isolated Storage Explorer allows you to see and explore files in the Isolated Storage. You can also import/export the file to/from the Isolated Storage.



WP7 Isolated Storage Explorer

- First let's download and install the Windows Phone 7 Isolated Storage Explorer installation file from CodePlex .
- Add the reference **IsolatedStorageExplorer.dll** to the project from

C:\Program Files\WP7 Isolated Storage Explorer\Library\IsolatedStorageExplorer.dll

- Start *IsolatedStorageExplorer* in the **App.xaml.cs** while launching.

```
private void Application_Launching(object sender, LaunchingEventArgs e)
    {
    IsolatedStorageExplorer.Explorer.Start("localhost");
    }
```

Localhost is the name/IP address of the system

- In *Activated* event in **App.xaml.cs** call `IsolatedStorageExplorer.Explorer.RestoreFromTombstone()`

```
private void Application_Activated(object sender, ActivatedEventArgs e)
    {
```


(C) Copyright Nokia 2014. All rights reserved.

```
    IsolatedStorageExplorer.Explorer.RestoreFromTombstone();
    }
```

- Run the application and launch the IsolatedStorageExplorer from

Start->All Programs-> WP7 Isolated Storage Explorer

- This will display the file being created, you can download the file to the desktop, delete the file or even can import file to the Isolated Storage.

Summary

When we create a file in Isolated Storage the I/O operations are restricted to, so the data are local to the application. It doesn't have direct access from any other file system or application. So it's secure from unauthorized access. If you are sharing same data from two applications then those data can't be local to any of the application, it suggested to store the data in cloud. The storage resources on a phone are limited, so it is recommended to store necessary data.

Source Code

The full source code of the example is available here: File:IsolatedStorageWP7.zip

(C) Copyright Nokia 2014. All rights reserved.