Command line tool to manage Windows 7 Libraries, with source code

devblogs.microsoft.com/oldnewthing/20120828-01

August 28, 2012

Raymond Chen

A customer asked if there was a command-line tool for managing Windows 7 Libraries so that they could create and deploy libraries across their organization.

Not only is there such a tool, it even comes with source code.

The Shell Library Command Line Sample shows how to manage Windows 7 libraries programmatically via the `IShellLibrary` interface. And it's actually a useful program on its own.

```
Usage: shlib.exe SUBCOMMAND
Displays and modifies the attributes of Shell Libraries.
Supported commands:
  create      Creates a library at the specified path.
  info        Prints info about the given library.
  enum        Enumerates the folders in the library.
  setattrib   Modifies the attributes of the library.
  add         Adds the specified folder to the specified library.
  remove      Removes the specified folder from the library.
  setsaveloc  Sets the default save location of the library.
  resolve     Resolves the specified folder in the library.
  resolveall  Resolves all locations in the library in bulk.
  manage      Displays the Manage Library Dialog for the library.
```

For example, to see all the folders that are part of your Documents library, use the command

```
shlib enum FOLDERID_DocumentsLibrary
```

Each of the commands has further sub-options.

```
Usage: create shlib.exe [OPTIONS]  [...]
Creates a library at the specified path.
Options:
 -create[:ARG]       Specifies that a new library should be created.
   <none>              Fail if the library already exists.
   overwrite           Overwrite any existing library.
   uniquename          Generate a unique name in case of conflict.
The library may be specified by a file system path,
or by a KNOWNFOLDERID (e.g. "FOLDERID_DocumentsLibrary").
```

And the source code shows how to perform each of these actions programmatically.

The `IShellLibrary` interface is also supported by the Windows API Code Pack for .NET, so you C# folks can use it too.

(One thing I noticed that is missing from the sample is a `seticon` command. Shouldn’t be hard to implement. You just call `IShellLibrary::SetIcon`.)

Raymond Chen

Follow