

How can I get information about the items in the Recycle Bin from script?

devblogs.microsoft.com/oldnewthing/20140421-00

April 21, 2014



Raymond Chen

Today we'll do a scripting version of an old C++ program: Printing information about the items in the Recycle Bin. (How you wish to act on the information is up to you.)

This is a pattern we've seen a lot. Bind to a folder, enumerate its contents, extract properties.

```
var shell = new ActiveXObject("Shell.Application");
var recycleBin = shell.Namespace(10); // CSIDL_BITBUCKET
var items = recycleBin.Items();
for (var i = 0; i < items.Count; i++) {
    var item = items.Item(i);
    WScript.StdOut.WriteLine(item.Name);
    WScript.StdOut.WriteLine(item.ExtendedProperty(
        "System.Recycle.DeletedFrom"));
    WScript.StdOut.WriteLine(item.ExtendedProperty(
        "System.Recycle.DateDeleted"));
    WScript.StdOut.WriteLine(item.Size);
}
```

Wow, that was way easier than doing it in C++!

Just for fun, I'll do it in C#, first as a straight port:

```

// add a reference to shell32.dll
class Program {
    public static void Main()
    {
        var shell = new Shell32.Shell();
        var recycleBin = shell.NameSpace(10); // CSIDL_BITBUCKET
        var items = recycleBin.Items();
        for (var i = 0; i < items.Count; i++) {
            var item = (Shell32.FolderItem2)items.Item(i);
            System.Console.WriteLine(item.Name);
            System.Console.WriteLine(item.ExtendedProperty(
                "System.Recycle.DeletedFrom"));
            System.Console.WriteLine(item.ExtendedProperty(
                "System.Recycle.DateDeleted"));
            System.Console.WriteLine(item.Size);
        }
    }
}

```

We have to cast to `Shell32.FolderItem2` because the default interface for the `Item()` method is `Shell32.FolderItem`, but `ExtendedProperty` is a method on `Shell32.FolderItem2`. We didn't have to do this explicit cast in JavaScript because JavaScript is a dynamically-typed language.

So let's use the `dynamic` keyword to mimic that in C#. Note, however, that if you use `dynamic`, then you miss out on a lot of IntelliSense features.

```

class Program {
    public static void Main()
    {
        var shell = new Shell32.Shell();
        var recycleBin = shell.NameSpace(10); // CSIDL_BITBUCKET
        var items = recycleBin.Items();
        foreach (dynamic item in items) {
            System.Console.WriteLine(item.Name);
            System.Console.WriteLine(item.ExtendedProperty(
                "System.Recycle.DeletedFrom"));
            System.Console.WriteLine(item.ExtendedProperty(
                "System.Recycle.DateDeleted"));
            System.Console.WriteLine(item.Size);
        }
    }
}

```

Now you can do things like list all the files deleted today

```
class Program {
    public static void Main()
    {
        var today = DateTime.Today;
        var shell = new Shell32.Shell();
        var recycleBin = shell.NameSpace(10); // CSIDL_BITBUCKET
        var items = recycleBin.Items();
        foreach (dynamic item in items) {
            if (item.ExtendedProperty("System.Recycle.DateDeleted").Date
                == today) {
                System.Console.WriteLine(item.name);
            }
        }
    }
}
```

Raymond Chen

Follow

