

## Folder Pwner

The Folder Pwner can be used to monitor file system activity in one or more specified folders and notify you accordingly. A simple xml file is used to set up how the Folder Pwner works. To use, make sure FolderPwner.exe and your configuration file (called config.xml by default) are in the same folder, and run the program. It will run in the background invisibly and you will have to use your Task Manager to close it.

Here is a sample configuration file:

```
<?xml version="1.0" encoding="utf-8" ?>
<config>
  <watcher>
    <path>c:\</path>
    <filter>*.txt</filter>
    <hook type="Created">
      <popup>File created: %name%</popup>
    </hook>
    <hook type="Changed">
      <popup>File modified: %name%</popup>
    </hook>
    <hook type="Deleted">
      <popup>File Deleted: %path%</popup>
    </hook>
    <hook type="Renamed">
      <system>ipconfig /all</system>
    </hook>
  </watcher>
</config>
```

Here is an example of each of the 4 types of hooks (names are case sensitive so pay attention) and both of the types of actions you can take when the file system is modified.

## **Commands**

Each hook node can contain multiple commands, and even multiple copies of the same type of command.

\*popup commands are simply a small window that pops up with a message and an “ok” box

\*system commands will run a system command just like you had typed it into your command prompt

## **Variable Keywords**

In any of the commands you can use variable keywords surrounded by %'s to inject certain values based on the file that was modified.

%name% - the current name of the file

%path% - the current full path of the file

(if it was a rename action then you have available these additional variables)

%oldName% - the original name of the file

%oldPath% - the original full path of the file