Computer Education

Tips Blog WinXP Internet Computing Downloads Vista Home

# Using INF Files to Edit the Windows Registry

**One way to edit the Windows Registry is with a special file type with the extension INF. Its construction and use is described.**

## Background and Introduction to INF Files

INF (Information) files have been around for a long time. They are text files with the file extension *.inf*, which can be used to ca out a variety of computer tasks. Most PC users will have encountered them as a way to install drivers. They also provide a method for editing the Registry and I sometimes used them for this purpose back in the days of Windows 95. However, in late systems I more or less forgot about INF files in favor of REG files or VBScripts.

INF files fell out of favor for Registry editing partly because their structure can seem a little arcane and is not explained in mar places. Even many computer professionals are not sure how to write an INF file or even interpret them. The main reference is the Windows Driver Development Kit, where the discussion of INF files is none too clear. Another Microsoft reference on INF architecture is here.

Nonetheless, INF files have advantages and, recently, I have returned to them as a way to edit the Registry. I have been partl motivated by the Windows Registry expert, Jerry Honeycutt, whose book, *Windows XP Registry Guide*, argues for using INF f and has a useful and clear explanation of their construction. Also, I think that INF files have advantages for use by average ho PC users. I summarize some of these next.

## Advantages of INF files

- Can delete individual Registry values whereas REG files can only delete entire keys
- Less likely to trigger anti-malware alarms
- Have bitwise support
- Have a built-in installation procedure in the standard right-click context menus
- Can include a procedure for uninstalling whatever they do
- Can be packaged in an EXE file with the Windows XP accessory, *Iexpress.exe*
- Files are easy to write once you understand the format

## Structure and Contents of INF Files for Registry Editing

INF files can have many sections that carry out a variety of functions but I will limit the discussion here to editing the Registry. will consider files that either add, change, or delete something in the Registry. (I will skip the case where bitwise arithmetic mi be involved since bitwise arithmetic can get tricky and will be left to some other time.). The sections of INF files are headed by names that must be enclosed in brackets. The simplest files have three sections. The first section tells the Windows operatinc system that the file is an INF file and is headed *[Version]*. The second section defines an action that is to be done if the "Instal entry in the right-click menu is used. The action (also called a *directive*) will be from a predefined set that the system recogniz as an INF function. In our discussion, the directives are *AddReg* for adding or replacing Registry entries and *DelReg* for deleti Registry entries. This second section is headed *[DefaultInstall]*. This section will also point to a third section that has the actua instructions for the action defined by the directive in the second section. The structure will become clearer with a few simple examples that I will show below. I will include comments within the sample files to help with the explanation, Note that comme are preceded by a semicolon. I will also color comments blue to distinguish them from actual content.

### First example of an INF file. Adds or replaces a Registry entry

```
[Version]
Signature=$CHICAGO$
;This somewhat mysterious entry identifies the file as an INF file. The CHICAGO reference is
relic dating back to Windows 95. Files can also be given the signature "$Windows NT$" to
indicate that the file is compatible with 32-bit operating systems only. This first section 
a requirement in INF files.

[DefaultInstall]
AddReg=Add.Settings
;This section defines what action is to be taken when the INF file is installed. There is a 
of predefined possible actions or directives that the setup API of Windows will recognize. I
this case, the directive is "AddReg", which is the directive to add a new Registry entry or
replace an existing one. More than one directive can be in this section if desired.
"Add.Settings" is the name of the section where the specific instructions of what to add are
be found. The name of this latter section is arbitrary but the name used here is common and 
descriptive.

[Add.Settings]
HKCU,Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced,HideFileExt,0x10001,0x00000
;These are the specifics indicating what is to be added or replaced in the Registry. The syn
is explained a bit further on in this article. This particular entry makes file extensions
visible for the current user.
```

### Second example of an INF file. Deletes a Registry entry

```
[Version]
Signature=$CHICAGO$

[DefaultInstall]
DelReg=Del.Settings
;The directive "DelReg" indicates that a Registry entry is to be deleted. The specific
instructions are indicated to be in the section headed "Del.Settings". The name of this last
section is arbitrary.

[Del.Settings]
HKCU,Software\HacksRUs\Example
;Removes a fictitious Registry key
```

### Third example of an INF file. Adds one Registry entry and deletes another

```
[Version]
Signature=$CHICAGO$

[DefaultInstall]
AddReg=Add.Settings
DelReg=Del.Settings
;There are two different directives this time. Instructions are indicated to be in two
different sections. Note that the system will always do a deletion first, irrespective of the
order of the directives inside the INF file.

[Add.Settings]
HKCU,Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced,HideFileExt,0x10001,0x000000

[Del.Settings]
HKCU,Software\HacksRUs\Example
```

If desired, an INF file can have numerous sections to carry out many Registry edits. A schematic example is shown next.

### Fourth example of an INF file. A more complicated version that edits several Registry entries

```
[Version]
Signature=$CHICAGO$

[DefaultInstall]
AddReg=Add.Settings1,Add.Settings2,Add.Settings3
DelReg=Del.Settings

[Add.Settings1]
;First Registry edit goes here
[Add.Settings2]
;Next Registry edit goes here.
[Add.Settings3]
;Next Registry edit goes here.
[Del.Settings]
;Next Registry edit. This one is a deletion.
```

## INF File Syntax for Adding or Replacing Registry Entries

The entry in an INF file that is used to add or modify existing Registry settings is written in parts, each separated by a comma The syntax for the code must be followed exactly. It is given by

```
    rootkey,[subkey],[value],[flags],[data]
```

Note that the full name of a key is broken into two parts separated by a comma. The first part is one of the major Registry key HKCR, HKCU, HKLM, or HKU. These abbreviations must be used. Table I gives the meaning of the abbreviations. In the example INF files given above, the rootkey is HKCU.

*Table I. Abbreviations for rootkeys*

| Abbreviation | Key |
|---|---|
| HKCR | HKEY_CLASSES_ROOT |
| HKCU | HKEY_CURRENT_USER |
| HKLM | HKEY_LOCAL_MACHINE |
| HKU | HKEY_USERS |

The remainder of the name of a key, the subkey, is written as one long string. In the first example above, the subkey is

```
Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced
```

If a subkey is omitted, actions will be taken on the rootkey.

The next part in is the *value* name. In the first INF example above, the value is *HideFileExt.* The value is followed by a flag tha indicates what type of Registry value it is. In the example, the flag *0x00010001* indicates that the value is a DWORD. Table II lists the common flags.

*Table II. Flags for value types*

| Flag | Value type |
|---|---|
| 0x00000000 | REG_SZ |
| 0x00000001 | REG_BINARY |
| 0x00010000 | REG_MULTI_SZ |
| 0x00020000 | REG_EXPAND_SZ |
| 0x00010001 | REG_DWORD |
| 0x00020001 | REG_NONE |

The final piece is the actual data for the value. In the example the data entry is *0x00000000*. The hexadecimal format is used because this particular example is a DWORD. Other value types would require the appropriate format. For example, the strinç type REG_SZ would have data of the form *"somestring"* where the quotation marks are part of the data.

To help keep the discussion clear, here is the complete example again:

```
HKCU,Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced,HideFileExt,0x10001,0x000000
```

## INF File Syntax for Deleting Registry Entries

The syntax for deletions is similar but is shorter. For our purposes it is given by

```
rootkey,[subkey],[value]
```

(Under some circumstances, there might also be a Flags and a Data part but that is beyond the scope of our discussion.). If tl *value* section is omitted, the entire subkey will be deleted.

## More Examples of INF Files for Editing the Registry

I have written some very simple INF files for carrying out useful Registry edits. Some examples are decribed below and can b downloaded by clicking the links. As usual, I must make the disclaimer that they are provided as is and the user is entirely responsible for any results.

- Make file extensions visible
- Hide file extensions
- Show "hidden" files
- Hide "hidden" files
- Clear "My Recent Documents" on exit

## How to install an INF file

The actual command that installs an INF file is

```
    RunDll32.exe setupapi,InstallHinfSection DefaultInstall 132 somefile.inf
```

However, unless you are writing a script, the usual way to install an INF file is to right-click the file and choose "Install" from th context menu. Normally, Registry edits will not take effect until after the user logs off and back on for HKCU edits or after a reboot for edits that affect all users.

<< Home page

This page last updated on: 01/17/2008 08:17:38

Home page >>