

본 컬럼에 대한 모든 저작권은 DevGuru에 있습니다.  
컬럼을 타 사이트 등에 기재 및 링크 또는 컬럼 내용을 인용 시 반드시 출처를 밝히셔야 합니다.  
컬럼 들을 CD나 기타 매체로 배포하고자 할 경우 DevGuru에 동의를 얻으셔야 합니다.

© DevGuru Corporation. All rights reserved

기타 자세한 질문 사항들은 웹 게시판이나 [support@devguru.co.kr](mailto:support@devguru.co.kr)으로 문의하기 바랍니다.

# INF (Information file) 파일에 대해

© 2003 Devguru ( Device driver Guru ), Inc.

## INF 파일이란 ?

드라이버를 제작하고나서, 마지막으로 해야 할 작업은 사용자에게 배포를 할수 있게 인스톨프로그램을 만들거나, 설치화일을 만드는 것이다. 그중에서 INF 파일을 작성하는 것은 가장 복잡하고, 난해한 내용을 가진 것중에 하나이다.

Microsoft® Windows Drivers 는 설치를 하기 위해 INF 파일이 필요하다. INF 파일은 셋업 구성요소에 의해 사용할 드라이버 파일, 레지스트리 정보, 버전 정보등등 설치할 디바이스와 파일에 대한 모든 필요한 정보를 포함하는 텍스트 파일이다.

## INF 파일의 구조

INF 파일은 섹션( Section ) 이라고 불리는 테이블들로 구성되어있다. 각각의 섹션은 이름을 가지고, [] 대괄호로 둘러싸여진다.

Ex ) [SectionName]

모든 섹션 이름은 대소문자 구분이 없이, 이론적으로 255 공백이 없는 ANSI or Unicode 문자를 가질수 있다. (대신, Win9x 에서는 19 ANSI 문자보다 적어야 한다.). 공백, 탭, 다른 컨트롤 문자들 [%];"W 은 허용이 안된다. (경우에 따라서는 될때도 있다 )

섹션 이름에 여러 개의 단어를 사용하기 위해서 “.” 를 사용하면 안된다. 왜냐하면 그 문자는 이미 Microsoft 에서 예약된것이기 때문이다. 대신 “\_” 또는 “- “ 를 사용하면된다.

Ex ) [My\_model-section]

각각의 섹션에는 디렉티브 ( directives )를 가진다. 그것은 섹션에 존재하지 않을수도 있고, 디렉티브 ( directives )는 무엇을 할지 시스템 디바이스 인스톨러 ( System device installer ) 에게 말한다. ( 파일 복사, 레지스트리에 엔트리 추가 등등 ). 주석문자는 세미콜론(;)이고, 그것은 줄에 아무곳에서나 사용할 수 있다. INF 에 사용하는 것에서 비교문장같은 것은 없다. “IF-THEN-ELSE” 나 다른 비슷한 구조를 가질수 없다. 그러나 섹션이름은 특정플랫폼에서 인스톨 할 때 구분을 할 수 있게 플랫폼에 맞는 접미사로 만들 수 있다.

Ex ) [SectionName.NT] - Windows 2000 이후 버전

위의 섹션은 Windows 2000 이나 그 이후의 윈도우에서만 작동하게 된다.

섹션들은 파일에서 보여지는 순서대로 실행되는 것이 아니라, 인스톨 과정에 맞게 실행된다. 그러므로, 섹션은 파일에 어느 곳이든지 나타날 수 있다. 대신 습관적으로, [version] 섹션은 항상 처음에 오고, [Strings] 섹션은 항상 마지막에 온다.

INF 파일에는 사용자에게 보여지는 여러 문자들이 있다. 그것들은 C 언어의 #define 매크로와 같이 문자열 대입할 수 있는 것이 있다. 문자열 대체 매크로는 [Strings] 섹션에 정의된다. 섹션에 디렉티브들은 문자열 매크로로 정의된다.

Ex ) [Strings]

```
my_company = "California Death Beams and Stuff"  
my_device = "Basselope 2003 Stealth Weapons System"
```

시스템 디바이스 인스톨러는 %매크로이름% 형태의 호출을 하는곳에서 찾은 위치에 매크로 대체를 수행한다.

Ex ) [Manufacturer]

```
%my_company% = install_my_stuff
```

일반적으로 컴퓨터를 사용하중에 영어권 뿐만 아니라 비영어권 사람들도 많이 사용한다. 그래서 컴퓨터 화면에도 영어뿐만 아니라 각각의 언어에 맞게 보여지길 원한다. 그래서 INF 파일에도 언어에 따라 보여지는 문자를 선택하게 할 수 있다. 그리고, 시스템 디바이스 인스톨러는 사용자가 시스템에 설치된 언어에 기반해서 올바른 문자열을 선택할 것이다. 여러분들은 여러 개의 [Strings] 섹션에 의해 이것을 정의 할 수 있다. 각각의 String 섹션은 Language ID 로 선언될수 있고, Language ID 는 SDK 에서 사용되는 것과 같은 형태를 가진다. 16 비트 중에 하위 10 비트는 주 언어를 나타내고, 상위 6 비트는 sublanguage 를 나타낸다. 예를들면, U.S English 는 0409 이다. Language 와 SubLanguage ID 의 정의는 SDK 에 있으니 그걸 참고하기 바란다. 예를들면, 다음과 같다.

```
[Strings] ; Default, US English
```

```
one = "One"
```

```
two = "Two"
```

```
[Strings.040c] French (Standard)
```

```
one = "Un"
```

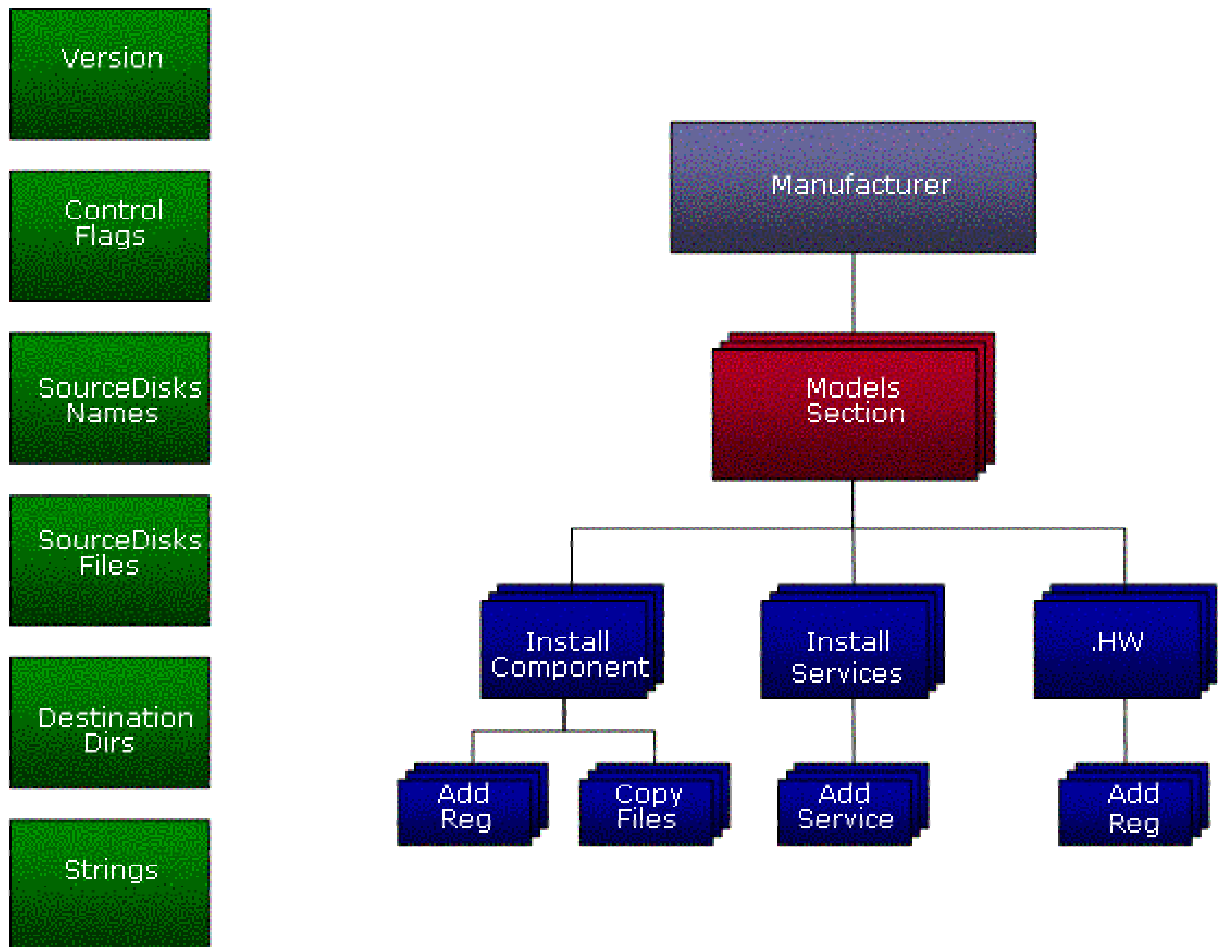
```
two = "Deux"
```

```
[Strings.080a]           ; Spanish (Mexico)
one = "Uno"
two = "Dos"
```

위에 [strings] 섹션에서, 시스템 디바이스 인스톨러가 INF 파일에서 %one% 또는 %two% 를 만나게 되면, 그 문자들은 [string] 섹션의 문자열로 대체될 것이다. 만약 사용자가 [string] 섹션에 지정되지 않은 다른 언어를 시스템에서 사용하고 있으면, 시스템은 기본 [string] 섹션을 선택할 것이다.

### 주로 사용되는 Sections 들

여러분들이 INF 파일에 대한 DDK 문서를 보면, 여러분들은 거기에 시스템 디바이스 인스톨러에 의해서 이해할 수 있는 39 개 정도의 섹션들이 있다. 그리고, 그 섹션들은 처음에 언급한것처럼, INF 파일의 섹션 대부분이 계층구조형태로 되어있다.



< 그림 1 >

<그림 1> 에 보여지는 그림처럼 왼쪽의 섹션들은 계층에 속한 부분이 아니고, INF 파일의 전체에서 필요한 부분이다. 계층에서 가장 맨 위 계층은 [Manufacturer] 섹션이다. 그리고, 섹션 이름들이 요구된다. 왜냐하면 시스템 디바이스 인스톨러에 의해 호출받기 때문이다. (Manufacturer, Strings, Version, .Service, .HW 등등) 다른 섹션이름은 여러분들이 작성한 INF 파일 내에서 참고되기 때문에 여러분들이 좋아하는 이름으로 작성할수 있다. 다음은 자주 사용되는 섹션과 그 섹션내에서의 디렉티브들에 대해 알아보자. 섹션의 디렉티브와 디렉티브들에 대한 내용은 DDK 문서를 참고하자.

## Version

이 섹션은 모든 INF 파일들의 기본적으로 사용되는 헤더 부분이다. 이곳에서는 O/S 와 Class Installer 가 이 INF 파일을 파싱할 기본적인 내용들을 작성한다.

Ex ) [Version]

```
Signature = "$Chicago$"           ; or $Windows NT$ for 2000/XP
Class = Test class name
ClassGUID = {30320101-C613-11d2-9647-0020AFEB03E0}; My GUID
Provider = %my_company%          ; My company
CatalogFile = MyCat.cat          ; WHQL checksum file
DriverVer = 04/01/2003,1.0.121.1 ; Driver date & version
```

*Signature* - 어느 O/S 가 이 INF 파일을 사용할 수 있는지를 지정한다. \$Chicago\$ ('Chicago 는 Windows 95 의 코드명이다. ) 는 윈도우의 모든 버전에서 사용할 수 있다는 것을 가르키고, \$Windows NT\$ 는 Windows 2000 과 그 이후의 버전에서 사용할수 있다는 것을 가르킨다.

*Class* - 디바이스가 속한 종류의 이름을 나타낸다. 미리 정의된 디바이스 종류와 GUID 들은 DevGUID.H 파일에서 찾아볼수 있다. 여기에서는 내가 클래스를 만들었다. Test class. If you create your own class name, then you must also have a ClassInstall32 section.

*ClassGUID* - INF 파일에 디바이스 class GUID 에 대한 내용을 지정한다. Class 와 ClassGUID 는 같아야 하고, 같지 않으면 인스톨을 실패한다. 만약 여러분이 새로 class 를 만들었다면, 새로운 GUID 는 SDK 나 DDK 에 있는 GUIDGEN 프로그램을 이용해서 만들어야 한다. 만약 여러분이 알고 있는 Class 를 사용한다면, DevGUID.h 에 GUID 를 찾아서 사용하면 된다.

*Provider* - INF 파일을 작성한 사람을 나타낸다. 일반적으로 이곳은 디바이스를 만든 회사가 될 것이다. 그러나 자주 "Microsoft" 가 되어 있는걸 볼 것이다. 이것은 MS 가 미리 벤더들을 위해 많은 INF 파일을 만들었기 때문이다.

*CatalogFile* - 드라이버와 INF 파일에 대해서 디지털 서명을 받은 파일을 지정한다. 이 파일은 디바이스/드라이버가 하드웨어 호환성 테스트에 합격해서 로고를 받은 후에 WHQL 로 부터 받는다. 개발하는 동안에는 catalog 파일은 없을 것이기 때문에. 이 부분은 생략 되거나 아무 것도 없는 것으로 나타낼 수 있다.

*DriverVer* - 드라이버가 만들어진 날짜와 버전을 나타낸다. 시스템 디바이스 인스톨러는 사용할 INF 파일이 어떤것인지를 선택하고, 찾기 위해서 이 정보를 사용한다.

### SourceDisksNames

이 섹션은 인스톨할 때 시스템으로 복사할 파일을 포함하고 있는 인스톨 디스크를 지정한다.

ex ) [SourceDisksNames]

1 = "Fishbowl Installation Disk #1"

2 = "Fishbowl Installation Disk #2"

SourceDisksNames 섹션에서 사용되는 디렉티브들은 다음과 같다.

**disk-id = disk-description[, [tag-file], [unused,path] [, flags]]**

*disk-id* - 간단히 디스크의 순서번호이다. 일반적으로 디스크는 1 부터 시작해서 하나씩 증가한다.

*disk-description* - 디스크를 나타내는 문자열이다. 시스템 디바이스 인스톨러가 시스템에 삽입할 디스크를 요구할 때, 사용자에게 보여진다.

*tag-file* - 그 디스크에서만 알고 있는 파일의 이름이다.

*Unused* - Windows 2000 이나 그 이후의 Windows 에서는 사용하지 않는다. Windows 9x 에서 사용된다. 자세한것은 Windows 98 DDK 를 살펴보면 된다.

*path* - 지정된 디스크에 있는 서브 디렉토리를 가르킨다.

*flags* - 파일이 있는 cabinet 파일의 이름을 지정할 tag-file 를 따른다. WinXP 와 그 이후에 버전에서, cabinet 파일이 사용되면, 그다음 tag-file 이름이 flags 필드를 뒤따른다.

### SourceDisksFiles

이 섹션은 인스톨 과정에서 시스템으로 복사할 모든 파일을 나타낸다.

ex ) [SourceDisksFiles]

```
Fish.hex = 1           ; on disk #1
Fishbowl.sys = 2       ; on disk #2
```

SourceDisksNames 섹션과 연관성을 가지고 실행된다.

디렉티브의 형식은 다음과 같다.

```
filename = disk-id[, [sub-directory][, size]]
```

*filename* - 인스톨 디스크에 있는 파일의 이름

*disk-id* - 파일을 포함하는 디스크이다. SourceDisksNames 섹션에서 선언되어 있다.

*sub-directory* - 파일이 위치한 디스크의 서브디렉토리를 나타낸다.

*size* - 바이트로 파일의 압축안된 크기이다. 일반적으로 사용하지 않는다.

### DestinationDirs

이 섹션은 파일을 복사할 시스템의 디렉토리를 지정한다. 파일과 디렉토리 번호의 리스트를 포함하는 섹션의 이름을 지정한다.

Ex ) [DestinationDirs]

```
Fb-W98-Driver-Files = 10, System32\Drivers
Fb-W2K-Driver-Files = 12
```

```
[Fb-W98-Driver-Files]
```

```
driver.sys
```

```
another_file.dat
```

```
[Fb-W2K-Driver-Files]
```

```
driver.sys
```

```
different_file.dat
```

이 섹션에서의 디렉티브를 위한 형식은 다음과 같다.

**section-name = directory-id[,sub-directory]**

*section-name* - INF 파일 만드는 사람이 만든 섹션 이름이다. 이 섹션에서는 특정 디렉토리로 복사할 파일이름의 리스트를 담고있다.

*directory-id* - 시스템 디렉토리를 위해 미리 정의된 식별자이다. 대부분의 주로 드라이버 작성자들을 위한 디렉토리 ID 들을 사용한다.

11 = %WinDir%\System32, 이고, 12 = %WinDir%\System32\Drivers, %WinDir% 는 string 섹션에 있는 매크로가 아니라, 여러분의 시스템에 미리 정의된 환경변수를 가르킨다.

*sub-directory* - *directory-id* 에 의해 지정된 서브 디렉토리이다.

### Strings

이 섹션에서는 INF 파일내에서 사용할 문자열 매크로 정의를 제공한다.

Ex ) [Strings]

```
ClassName = "Fishbowl"  
MfgName = "Sannas Consulting, LLC."  
Fishbowl = "Fishbowl USB Demonstration Device"  
ServiceDesc = "Fishbowl driver"
```

이 섹션의 포맷은 다음과 같다.

**macro-name = "some string"**

*macro-name* - 문자열을 위해 정의한 매크로 이름이다. 퍼센트(%) 문자로 둘러싸여서 INF 파일의 여러 곳에서 참조해서 사용한다.

예) %macro-name%.

"*some string*" - 매크로 정의내용이다. 이것은 INF 파일에서 보여지는 매크로의 내용을 이 문자열로 대체할것이다.

### Manufacturer



이 섹션은 디렉토리와 매우 비슷하다. 그것들은 디바이스가 INF 파일에 의해 지원되는 모든 회사들을 보여준다. 외부 회사를 위해, 하나 이상의 회사가 거기에 있는 것은 드물다. 그러나, 마이크로소프트에서 만든 INF 파일은 하나의 INF 파일안에 서술되어 있는 회사들이 여러 개 있다. Manufacture 섹션은 나머지 섹션을 위한 계층의 맨 위에 있다. ( 그림 1 참조 )

ex ) [Manufacturer]

```
%MfgName% = Sannas
```

이 섹션을 위한 문법은 INF 파일의 다른 섹션과는 다르다. 만약 여러분이 디렉티브를 보면, 큰 따옴표로 되어 있는 문장이 ( 매크로로 대체되기도 함 ) 식별자로 사용하고 있는 것을 볼 수 있다. 그리고, 등호 기호는 할당하는 연산자로 사용되지 않는다. 대신 분리자로서 사용된다. 이 섹션은 시스템 디바이스 인스톨러가 인스톨과정 처음에 Hardware id 와 Compatible ID 와 같은 INF 를 찾는데 사용된다.

이 섹션은 형식은 다음과 같다.

**manufacturer-name = models-section-name**

*manufacturer-name* - 회사를 나타내는 문자열이다.

*models-section-name* - 이 INF 파일로 지원하는 회사들의 디바이스 모델들의 리스트를 포함하는 섹션이름이다. (이름은 INF 파일 작성자가 만든다.)

### Models

이 섹션은 INF 파일에서 지원하는 회사들의 다양한 디바이스 모델들을 나타낸다. 이 섹션의 이름은 INF 파일 작성자에 의해 결정되고, Manufacture 섹션에서 참조한다.

Ex ) [our\_devices]

```
%device1-name% = product-install, hardware_ID, hardware_ID
```

```
%device2-name% = product-install, compatible_ID
```

이 섹션의 형식은 다음과 같다.

**device-description = install-section-name, PNP-id[,PNP-id]...**

*device-description* - 디바이스를 설명하는 문자열 (주로 매크로로 나타낸다.)

*install-section-name* - INF 파일 작성자가 만든 섹션이름이다. 디바이스를 위한 드라이버를 설치할 내용을 담은 디렉티브이다.

*PNP-id* - hardware ID 와 compatible ID 이다 각각의 장치에 고유한 값이다.

다음의 예제를 통해 어떻게 INF 파일을 O/S 가 해석하는지 알아보자.

먼저, 시스템 디바이스 인스톨러가 특정 디바이스를 위한 INF 파일을 찾을 때 Manufacture 섹션부터 시작할 것이다. 그리고, Model 섹션을 찾을 것이다.

```
[California_Death_Beams],
```

이 INF 파일에 의해 지원된 회사로부터 디바이스의 목록을 포함한다.

만약 [California\_Death\_Beams] model 섹션에서 인스톨러가 찾는 hardware IDs 나 compatible IDs 가 같지 않으면, 그것들은 Manufacture 섹션의 다음 엔트리를 찾을 것이다.

그리고, [Texas\_Techo\_Gizmo] models 섹션을 위한 INF 파일을 찾을 것이다. 그리고, 리스트에서 hardware IDs 와 compatible IDs 들을 조사할것이다. 만약, 같은 것을 찾으면, 인스톨러는 이 INF 파일이 디바이스 를 지원하는것이라고 결정하고, 지정된 install-section-name 은 디바이스를 위한 드라이버를 설치 한다.

```
[Manufacturer]
```

```
%cdfs% = California_Death_Beams
```

```
%ttg% = Texas_Techno_Gizmo
```

```
[California_Death_Beams]
```

```
%phaser% = install-phaser, pciWven_9876&dev_1234
```

```
%blaster% = install-blaster, pciWven_9876%dev_5678
```

```
[Texas_Techno_Gizmo]
```

```
%bass% = install-bass, dogWtype_basselope&model_2003
```

```
[install-phaser]
```

```
[install-blaster]
```

```
[install-bass]
```

```
[install-bass.NT]
```

```
[strings]
```

```
cdb = "California Death Beams and Stuff, Inc."
ttg = "Texas Techo Gizmo, LLC."
phaser = "Hand-held phaser, model 1234"
blaster = "Hand-held blaster, model 5678"
bass = "Basselope 2003 Stealth Weapons Platform"
```

### InstallComponent

이 섹션 ( DDK 문서에서는 DDInstallSection 라고 하기도 한다 ) 은 디바이스를 위한 드라이버를 인스톨하기 위해 요구되는 동작을 수행한다. 이 섹션의 이름은 INF 파일 작성자에 의해 결정된다. 그리고, 이 섹션은 Models 섹션에서 참조된다.

```
ex ) [Fishbowl]
```

```
CopyFiles = Fb-W98-Driver-Files
AddReg = AddReg-Win98, AddReg-All
```

```
[Fb-W98-Driver-Files]
```

```
Fishbowl.sys
Fish.hex
```

```
[AddReg-Win98]
```

```
HKR,,DevLoader,,*ntkern
HKR,,NTMPDriver,,Fishbowl.sys
```

```
[AddReg-All]
```

```
HKR,,DebugLevel,0x00010001,2
HKR,,FirmwareFile,0x00000000,\\SystemRoot\\System32\\Drivers\\Fish.hex
HKR,,AutoloadFirmware,0x00010001,1
```

이 섹션은 많은 디렉티브들을 지원한다. 대부분 CopyFiles 과 AddReg 이다.

*CopyFiles* - 파일 이름의 리스트를 포함하는 INF 파일 작성자에 복사할 파일의 이름을 지정한다.  
( 전에 SourceDisksFiles 섹션에서 지정 )

AddReg - 레지스트리에 추가할 내용을 포함하는 섹션의 이름 ( INF 파일 작성자에 의해 만든다 ) 을 포함한다.

### .Services

InstallComponent 섹션이 실행된 후에, 시스템 디바이스 인스톨러는 INF 파일에서 InstallComponent 섹션과 동일한 섹션 이름, “.Services” 로 끝나는 섹션을 찾는다. .Services 섹션은 서비스 컨트롤 매니저에게 드라이버를 설명하는, 레지스트리의 “services” 키에 엔트리를 추가할 때 사용된다.

```
[Fishbowl.NT.Services]
```

```
AddService = Fishbowl, 0x00000002, Add-Service
```

```
[Add-Service]
```

```
DisplayName = %ServiceDesc%
```

```
ServiceType = 1 ; SERVICE_KERNEL_DRIVER
```

```
StartType = 3 ; SERVICE_DEMAND
```

```
ErrorControl = 1 ; SERVICE_ERROR_NORMAL
```

```
ServiceBinary = %11%WDrivers\Fishbowl.sys
```

이 섹션에서는 여러 개의 디렉티브들을 사용한다. 그러나 대부분은 AddService 를 사용한다.

*AddService* - 드라이버를 설명하는 디렉티브들을 포함하는 INF 파일 작성자에 의해 만들어진 섹션의 이름을 지정한다.

### .HW

이 섹션은 주로 PnP 필터 드라이버를 설치하는데 사용된다. .Services 섹션이 실행된후에, 시스템 디바이스 인스톨러는 InstallComponent 섹션과 동일한 이름을 가진 섹션, “.HW” 로 끝나는 이름을 INF 파일에서 찾는다..HW section 은 주로 “hardware” 키에 엔트리를 추가할 때 사용한다.

```
(HKLM\System\CCSW\Enum\<enumerator>\<device-id>\<instance-id>).
```

```
[Snoopy.NT.HW]
```

```
AddReg = AddReg-HW
```

```
[AddReg-HW]
```

```
HKR,, "UpperFilters", 0x00010000, "Snoopy"
```

이 섹션은 여러 개의 다른 디렉티브를 지원한다. 그러나 대부분은 AddReg 이다. 레지스트리에 추가할 엔트리를 포함하는 섹션을 지정한다. upper filter ( F00 의 위에 ) 로 로딩하기 위해서 "UpperFilters" 키를 지정한다. 그리고, lower filter ( F00 의 밑에 ) 로 로딩되기 위해서 "LowerFilters" 키를 지정한다.

지금까지 INF 파일 작성에 사용되는 섹션들에 대해 알아보았다.

### 맺음말

드라이버를 사용자에게 배포하기 필요한 INF 작성에 대해 설명했습니다. 드라이버를 만들고 나서 설치를 어떻게 할지 몰라서 어려워하는 개발자들에게 도움이 되었으면 합니다. 다음 칼럼에서는 INF 파일이외에 다른 드라이버 설치 관련 내용들을 다뤄보도록 하겠다.