

1、驱动是什么？

(1) 基本境界，驱动就是一个安装包

最初很多人都是这样认为的，至少我是，但不乏很多干装机干了许多年的人还是这样认为的，例如我们这科技市场的一个“高手”，干了 7 年的装机还是认为驱动就是一个安装包，运行一下就行了。

处于这种境界时一般会按照硬件的类型，至少是按照硬件的“牌子”（例如 ATI 或 nVIDIA 的显卡，至于什么型号有些人并不一定知道）去搜寻驱动安装包，得到安装包后双击开始安装，完事。

不过我并不是在否定或贬低这种认识，这种认识并非是错误的，厂商提供驱动安装包也是为了最大简化驱动安装复杂度、提高驱动安装稳定性而做的，基本境界对驱动的理解和硬件厂商制作驱动安装包的目的是吻合的。

(2) 进阶境界，驱动就是一堆散乱的以 SYS、DLL、EXE、CAT 等为后缀的文件

一般人在手动使用 Windows 提供驱动搜索与安装程序进行驱动搜索与安装至少一次后，即会对驱动产生上述的理解。

在这种境界下，渐渐的认识到了驱动并非像普通软件那样，只能采取固定的安装方式。也慢慢的认识到，驱动的安装是系统提供的一种基础功能，这个基础功能不随驱动安装包的类型、硬件的类型不同而改变。

达到这种境界的人有很多，基本上大多数的人都处在这一层。不过处于这一层的人也分化为两类，一类人认为与驱动相关的所有的 SYS、DLL、EXE、CAT 等都是驱动必须的，至少绝大多数是必须的；另一类人则对每个 SYS、DLL、EXE、CAT 等文件究竟是什么作用有很有兴趣。

一般而言，此层次第一类人在用别人的驱动包也会自己做一些驱动包，而第二类人在用自己的驱动包或在研究比较牛的驱动包。

(3) 高级境界，驱动文件与驱动 INF 有关，且并非所有的文件都真的是“驱动文件”

一般而言，只有上一层次中的第二类人会达到这个境界。在长期的驱动搜集与研究工作中，在实践达到一定程度后，逐渐的会对哪些文件是真正关于驱动的、哪些文件是很少用到的、哪些文件非驱动文件但却是必须的有一定感悟。

达到这个境界的人，逐渐会开始对驱动文件与驱动 INF 的关系有一定的认识，会尝试去掉某些很少用的或基本无用的文件，会尝试简缩驱动的总体体积等，甚至某些人可以对 INF 进行一定程度的修改与定制。

这种境界是很有成就感的，达到这种境界后就像封装系统成功后的那种成就感一样，只不过以前是在定制系统，而现在是在定制驱动。顺道说一句，驱动里也可以加你的 OEM 信息，嘿嘿。

(4) 达人境界，驱动 INF 自己写，驱动文件自己改

达到这种境界的人相当少，对 INF 文件理解到一定程度后可以完全自己写 INF，还可以根据自己驱动的应用，加强驱动中的某些部分，例如现在某些改版驱动就会加强 OPENGL 在游戏中的效果。

这可能是玩驱动的人的最高境界了，不过或许也不是可望不可及的吧，希望有一天我们也能达到这个境界！

2、哪些文件才是真正的“驱动文件”

如果我们对一个驱动安装包进行解压，或者提取已经安装好的驱动，我们会发现“驱动”一般由如下几类文件组成：

(1) CAT 文件，这种文件用于驱动的微软认证，所谓微软认证就是经过微软测试并认可的驱动，相对而言，通过了微软认证的驱动具有较高的稳定性与可用性，但并非未通过认证的驱动就一定是不稳定、不可用的。但不要试图指望在修改驱动后 CAT 文件依旧能帮你通过驱动认证，之所以称作认证，是严格的、不可修改的。

(2) SYS 文件，一般的，SYS 文件就是真正的驱动文件，至少是诸多必须文件中的一个。我们每安装一个驱动都会在系统中创建一个驱动服务，每次我们进系统的时候这个驱动服务就会加载与这个服务相对应的 SYS 文件启动，从而“驱动”起硬件。最原始的驱动，就我个人理解，应该就是 SYS 文件以及与其对应的 INF 文件。

(3) EXE 文件，一般的，EXE 文件为控制器文件或者驱动相关服务所需的文件。控制器，如显卡的调整控制器，用以调整显卡的诸多性能与项目。驱动相关服务，如某些帮助服务、OPENGL 调节服务等。EXE 文件总体来讲用于各种调整功能，而 EXE 文件一般而言并非驱动所必须的文件。

(4) DLL 文件，DLL 文件被称为“动态链接库文件”，DLL 中一般包含其他文件要调用的函数或者功能，总之可以把 DLL 理解为一个仓库，其他文件执行时会在其中找到所需的。DLL 文件一般为控制器文件的扩展，或者为控制面板的各个功能。一般来讲，DLL 文件与 EXE 文件的关系比较密切，所以 DLL 也并非驱动所必须的文件，不过不排除例外，例如很多显卡驱动中的 DLL 也有其特殊的存在意义。

(5) CHM 和 HLP 文件，这两类文件中一般是驱动说明和使用帮助，大致上可以理解为说明书。既然是说明书，也就肯定与驱动本身无关，只是起扩展的解释说明作用。所以基本上而言，CHM 和 HLP 文件非驱动所需。

(6) INF 文件，INF 文件可以理解为一份驱动安装所需的配置文件。我们让系统搜寻并安装驱动时，系统就是根据 INF 文件的内容来决定：要复制哪些文件到什么位置、共需要哪些文件、有哪些设备是可以安装这个驱动的、驱动安装后显示什么名字、要为此驱动安装什么样的驱动服务等等。可以说 INF 文件决定着驱动的安装方式，起着指向作用。

3、玩驱动的目的何在？

玩系统的最终层次，在于玩驱动和玩编程——当然，这是对我们大多数非专业从事驱动与编程事业的人而言的，其实驱动说到底也是编程。

有人热衷于小玩意，例如改个 IE 图标、减少滚动条次数、改个桌面加个自己的名字、改个 OEM 信息、装个一键还原、装 3 个杀毒 2 个防火墙（……），当然，这些东西是必须要会的，但绝对不是会了这些就可以说自己能驾驭系统、驾驭驱动、驾驭计算机了。

要驾驭系统，其一就是要驾驭驱动。

为什么我一直在强调我们是在玩，一是因为我们一直是在改别人的成品，从系统到驱动，虽然没有一样是我们创造的，但是我们无一不把他们改的更有意思了；二是因为“玩”是一种良好心态，如果把一切当成任务，那么必然会失去最直接的动力——兴趣。

好吧，言归正传。

驱动是一个贯穿整个 Windows 环境下的东西，无论我们现在是在玩 XP 的驱动，还是将来我们会玩 Vista 甚至 Win7 的驱动，至少在将来的 10 年里，在 Windows 的桌面级操作系统里，驱动这个概念是不会有太大变化的。XP 活了 8 年，再活 2 年没问题，现在还有人在用 2K；Vista 的研发花了大价钱，就算再不济，微软也不可能让它倒的这么快；既然 Win7 也是桌面而非网络操作系统，那么 Windows 的这个模式还会存在至少 10 年。

虽然没有万用的东西，我们不可能让 XP 的驱动在 Vista 下有效，但是东西永远是死的，人永远是活的，只有方法、思想和精神是通用的。我们研究驱动的目的，在于掌握一种可以在 Windows 下通用至少 10 年的方法与思想。玩驱动给我们这些从事装机、维护等行业的人来说，最直接的效果是我们可以按我们的需要来修改、精简驱动。例如，修改的一个作用是可以让设备管理器里直接显示我们的 OEM 信息，精简的最直接作用是可以有效减少驱动的体积。但是，万万不可只注重修改与精简带来的最终目的，而忽略了过程中的东西。就像没认真玩过 nLite 的人对 XP 的结构组成绝对没有玩过 nLite 的人了解一样，精简与修改的意义不在于最后我可以改的多么花哨或是精简的像个骨头，而在于我们在做这些时，对系统、对驱动认识的逐步增加。

说到底，玩驱动会让我们多掌握一种方法，多理解一种事物。

说了这么多，是否让大家对驱动有了更多的理解呢？说这么多的目的，在于让大家正视驱动以及驱动这门技术。驱动不是装上就行的东西，就像我们有安装版 XP 还为什么研究封装部署 XP 一样，我们要用自己的知识与技术让驱动更好的为我们服务！