

# UEFI+GPT、Legacy+MBR 引导模式介绍及引导修复教程

——@南木嘉人

贴吧地址: <http://tieba.baidu.com/p/3754146793>

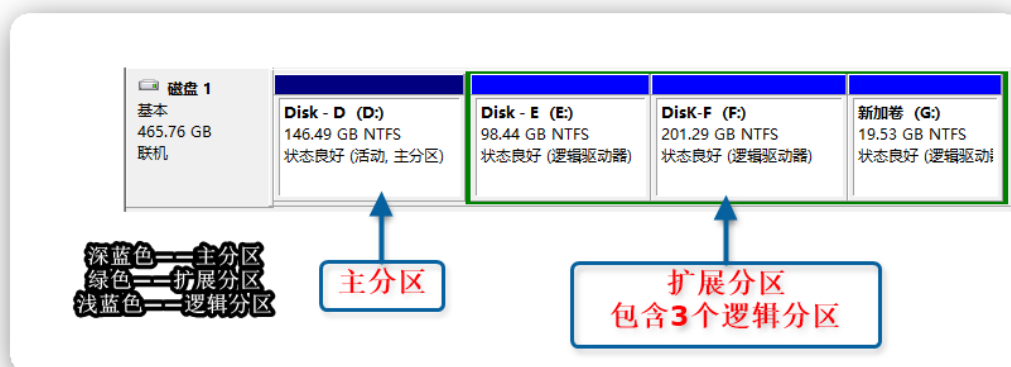
## 磁盘分区表格式

一般来说, 磁盘分区表有两种格式: **MBR 和 GPT**

**MBR 分区表**在 **windows** 操作系统下最多支持 4 个主分区或 3 个主分区+1 个扩展分区 (包含多个逻辑分区), 扩展分区必须划分为逻辑分区才能使用, 1 个扩展分区可以划分多个逻辑分区

MBR 分区表不支持容量大于 2.2TB 的分区(一些硬盘制造商将他们的容量较大的磁盘升级到了 4KB 的扇区,这意味着 MBR 的有效容量上限提升到了 16 TB)

如下图是一个 MBR 分区表示例: 1 个主分区+1 个扩展分区 (划分了 3 个逻辑分区)



**GPT 分区表**对分区数量没有限制, 但在 **windows** 系统上最多可以支持 128 个主分区

GPT 分区表突破了 MBR 最大支持 2.2T 分区的限制, 貌似最大支持 18EB 的分区

如下图是一个 GPT 分区表示例: 划分了 7 个主分区



## BIOS 引导方式

目前主要的系统引导方式也有两种: 传统的 **Legacy BIOS** 和新型的 **UEFI BIOS**

一般来说，有如下两种引导+磁盘分区表组合方式：**Legacy BIOS+MBR 和 UEFI BIOS+GPT**  
Legacy BIOS 无法识别 GPT 分区表格式，所以也就没有 Legacy BIOS+GPT 组合方式；  
UEFI BIOS 可同时识别 MBR 分区和 GPT 分区，所以 UEFI 下，MBR 和 GPT 磁盘都可用于启动操作系统。不过由于微软限制，UEFI 下使用 Windows 安装程序安装操作系统是只能将系统安装在 GPT 磁盘中。

✚ 再来说说传统 Legacy BIOS 和新型 UEFI BIOS 引导方式的工作原理吧：

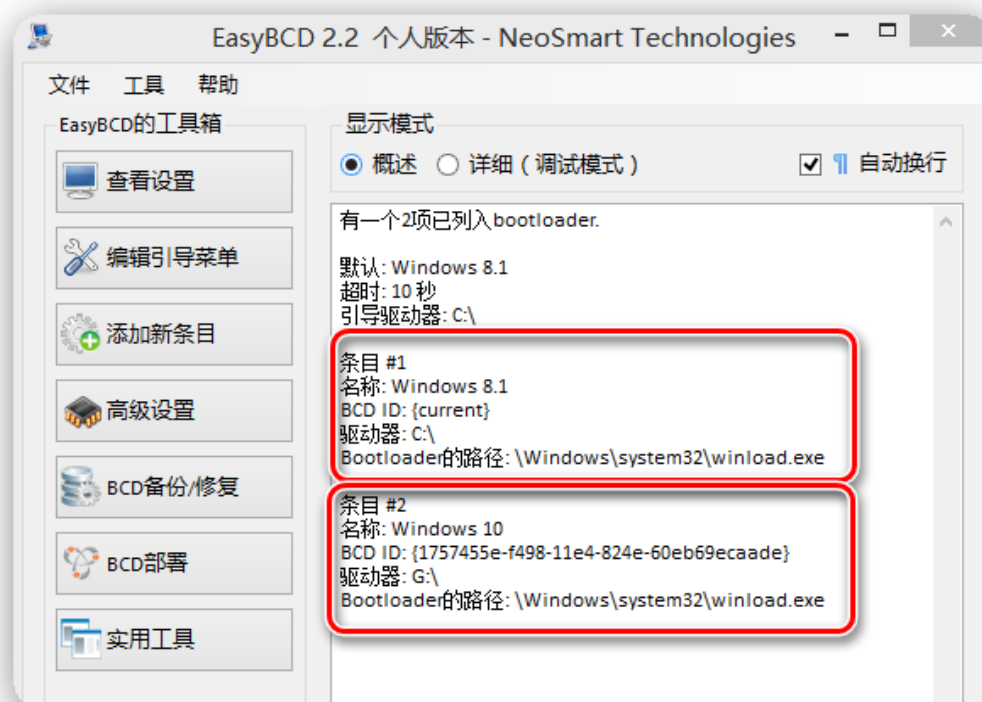
## Legacy BIOS

Legacy BIOS → MBR → “活动的主分区” → \bootmgr → \Boot\BCD → \Windows\system32\winload.exe

传统 Legacy BIOS 引导 windows 操作系统时，是通过一个**活动的主分区**下的 **bootmgr**（启动管理器）文件导入根目录下 boot 文件夹里的 **BCD**（启动设置数据）文件，然后 BCD 文件根据自身的配置内容加载系统启动文件 **winload.exe**（位置：根目录 \Windows\system32\winload.exe）来启动系统。

一个 BCD 文件可以加载多个系统启动文件从而实现引导多个系统的启动

通过 EasyBCD 工具可以看到 BCD 文件的内容，如下是我的 win8.1 和 win10 两个系统的 BCD 内容：

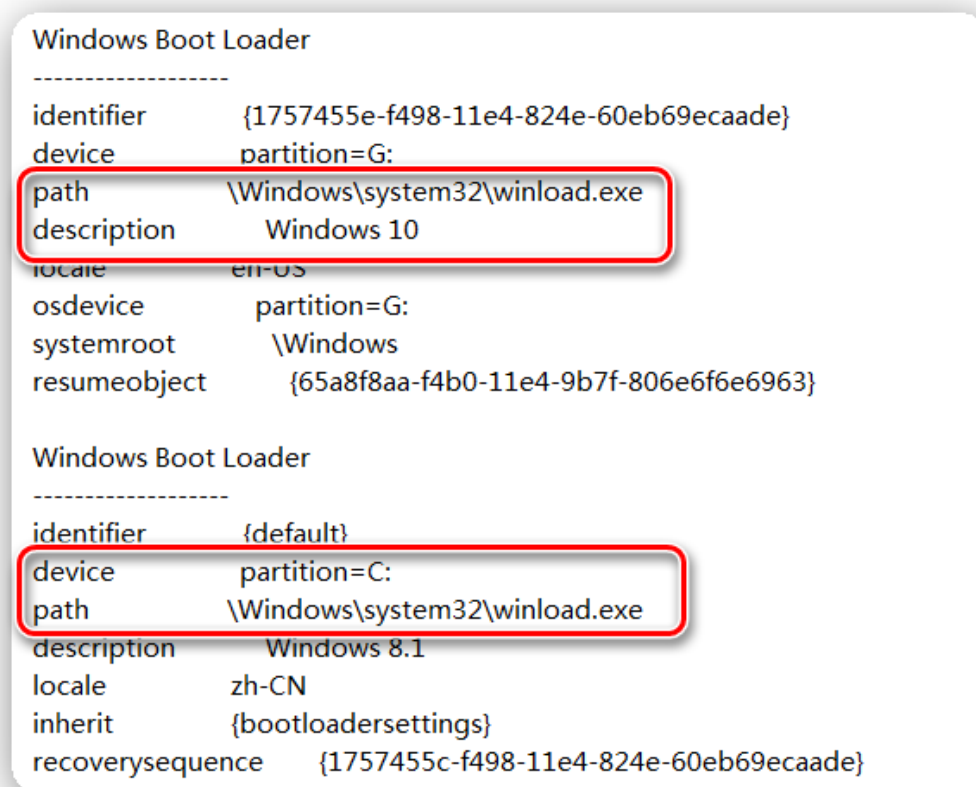


我的 windows 8.1 在 C 盘，winload.exe 位置：**C:** \Windows\system32\winload.exe

我的 window 10 在 G 盘，winload 位置：**G:** \Windows\system32\winload.exe

通过 BCD 来加载不同位置的 winload.exe 来实现这两个系统的启动

通过引导修复工具 NTBOOT Autofix 查看 BCD 文件内容，如下图：



需要注意的是：MBR 磁盘格式下，windows 系统的启动文件（(bootmgr、BCD)）必须存放在**活动的主分区**内，这样才能正常引导系统启动（**MBR 磁盘分区格式下，只允许有一个分区是活动的**）。

如果是使用微软原版操作系统按照 windows 安装程序来进行系统安装，系统会自动创建一个隐藏的活动主分区（win7 貌似是 100M，win8 貌似是 350M）用来存放启动文件；如果采用其他安装方式来安装系统，由于系统默认并不会自动创建这个活动的主分区，启动文件将会存放在我们的系统盘里，所以我们在采用其它安装方式安装第一个系统时，需要确保安装系统的分区**是活动的、而且是主分区**，而安装第二个、第三个...系统时，就不必要求必须是主分区了，逻辑分区也可以，因为安装第二个、第三个...系统时，我们已经有了一个活动的主分区了（第一个系统所在的分区），这个活动的主分区下的 BCD 文件里已经包含了我们的第二个、第三个...系统的启动信息用来启动第二个、第三个...系统。

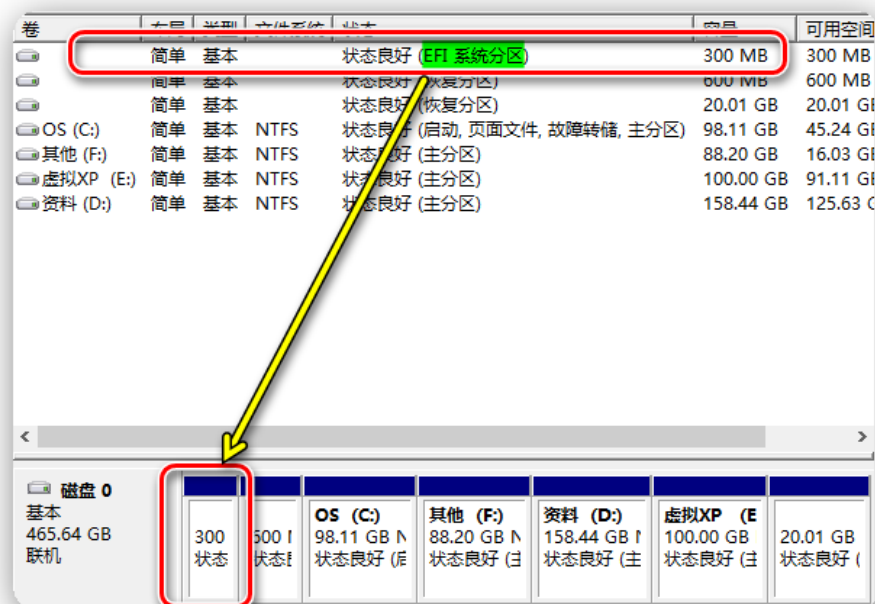
## UEFI BIOS

UEFI BIOS → EFI 系统分区（FAT 格式的分区）→ \efi\Microsoft\boot\bootmgfw.efi → \efi\Microsoft\BCD → \Windows\system32\winload.efi

UEFI BIOS 引导 windows 系统时，是通过一个 FAT 格式分区下的 bootmgfw.efi 文件来导入 BCD 文件，然后 BCD 文件根据自身的配置内容加载系统引导文件 **winload.efi**（对比 legacy 引导发现，UEFI 的引导文件为 winload.efi，而 Legacy 的引导文件为 winload.exe）

需要注意的是：GPT 磁盘格式下，windows 系统的启动文件（bootmgfw.efi、BCD）是存放在一个 FAT 格式的分区里的，有些出厂预装 win8 系统的电脑下将该 FAT 分区称之为 ESP 分区或 EFI 分区

如下图，ESP 和 EFI 分区一般都是隐藏的 FAT 分区，可以通过 DG 分区工具来创建 ESP 分区，还可以用 DG 为该分区添加盘符让其在资源管理器下显现出来



可见，UEFI+GPT 模式引导 windows 系统时，并不需要 mbr 主引导记录，也不需要活动分区，只需要你一个存放了引导启动文件的 fat 格式分区就可以了，这个 fat 分区当然也可以是 U 盘等外接 USB 设备了。

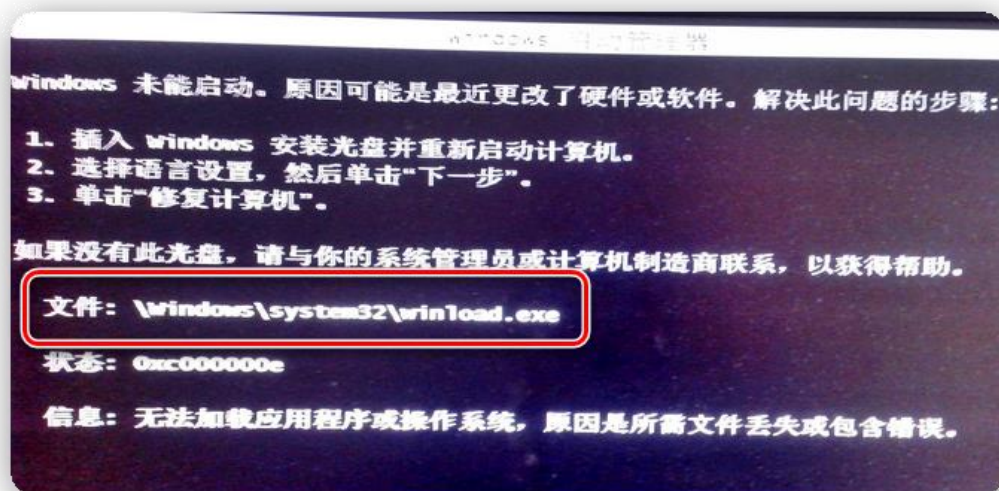
就目前情况而言，GPT 分区表磁盘不支持 32 位的 win7 以及 win7 之前的系统，支持 64 位的 XP、win7、win8、win10 和 32 位的 win8、win10。一般地，GPT 磁盘多与 64 位 windows 系统组合搭配。

## 引导修复

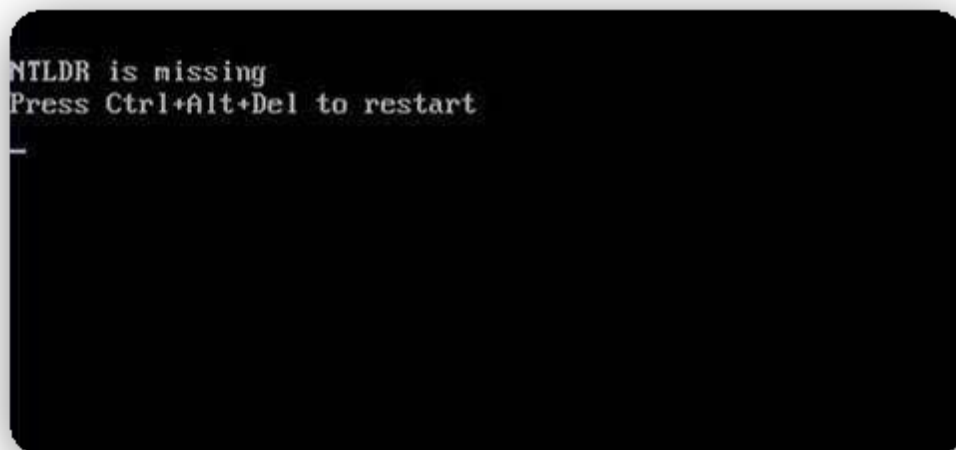
再来说说引导丢失、损坏导致系统无法正常进入情况下，如何通过修复引导来使系统正常启动。

### 引导问题故障举例

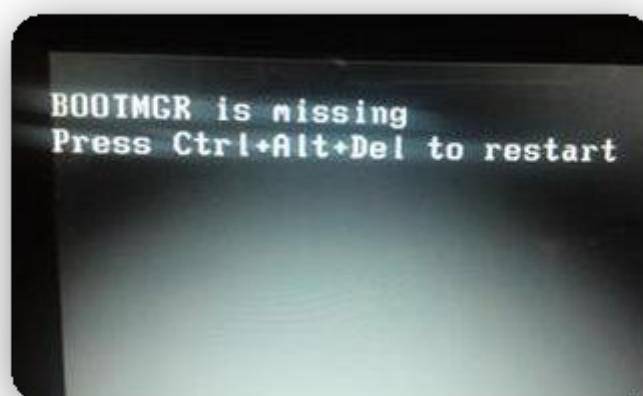
常见的引导丢失、损坏情况说明如下：



上图，Winload.exe 文件有问题，可见是 Legacy BIOS 引导文件出错；如果此处是 winload.efi，则应推测是 UEFI BIOS 引导文件出错。

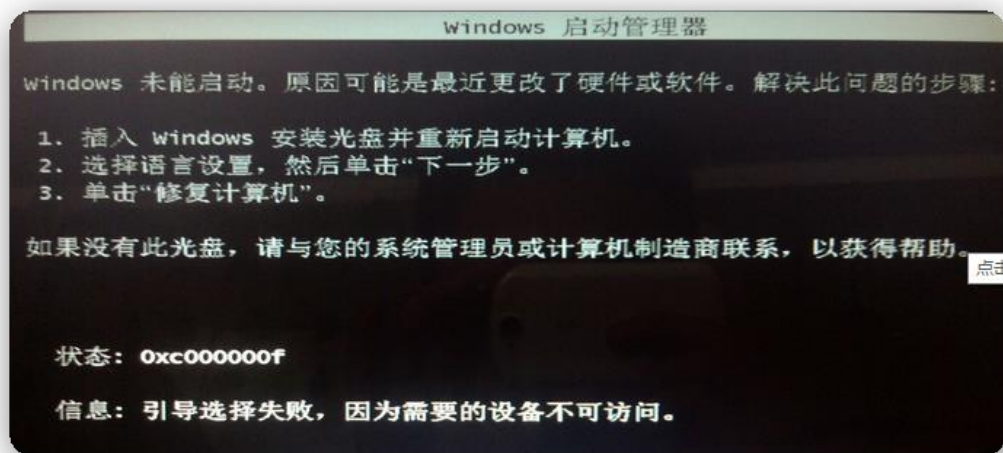


上图，NTLDR is missing，NTLDR 文件丢失。推断为:XP 等 NT5.x 架构操作系统引导丢失。  
**NTLDR** 是如 win 2000、XP、win 2003 等 NT5.x 架构操作系统的启动管理器文件，与之对应的 **bootmgr** 则是如 Vista、win7/8/9/2008/2012 等 NT6.x 架构操作系统的启动管理器，如下图：



上图，Bootmgr is missing，推断为:win7、win8 等 NT6.x 架构操作系统引导丢失。





## 引导修复

针对于以上各种引导问题给出引导修复解决方法

一般地，引导修复有很多种方法，牛逼的大神直接使用 dos 命令来搞定，那么针对于我们普通用户，我推荐你先使用网络上的引导修复工具来进行修复，等到使用熟练了，想多学习些技术，这时候再去研究 dos 命令吧。

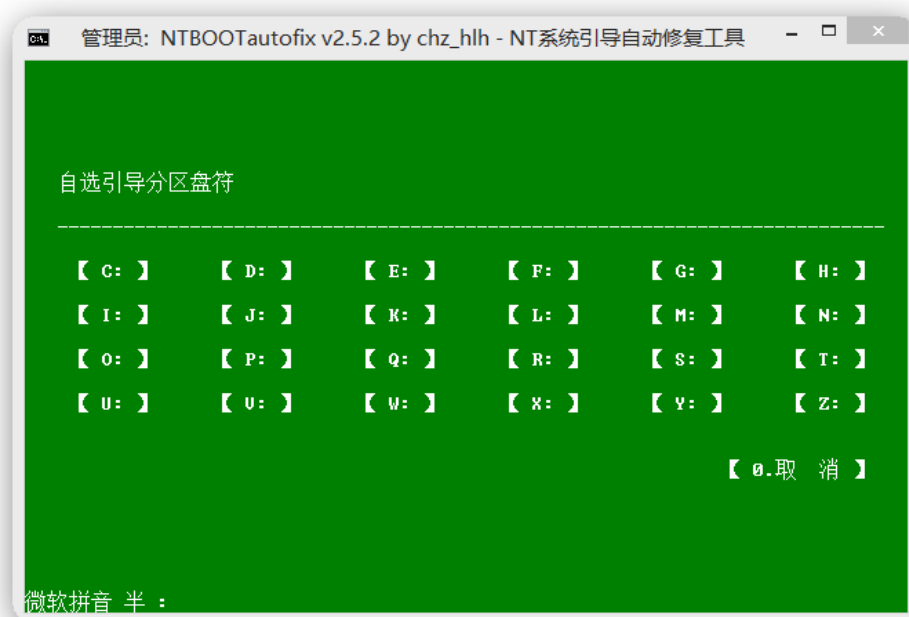
下面介绍的两种方法所涉及到 **NABOOT Autofix**、**BOOTICE**、**DG** 等工具一般 PE 下都会集成的，这里就不给大家放单独的下载链接了。PE 制作及使用的话，这里就不说了，记得 N 久 N 久 N 久以前就发过专门的教程了，如果你非要问我关于 PE 的东西，请百度“电脑店”“大白菜”“老毛桃”等关键字。

### 方法一：使用 NTBOOT AutoFix 工具来修复引导

引导修复工具有很多，这里推荐这款工具给大家，上手简单，操作简单

如果你的系统无法正常进入，那么请到 PE 下运行 NTBOOT AutoFix 进行修复；如果你是多系统，其中有一个系统可以正常进入，其它系统引导丢失，那么就可以在这个正常的系统下使用 NTBOOT AutoFix 进行修复，一般 PE 下都会集成这个软件。

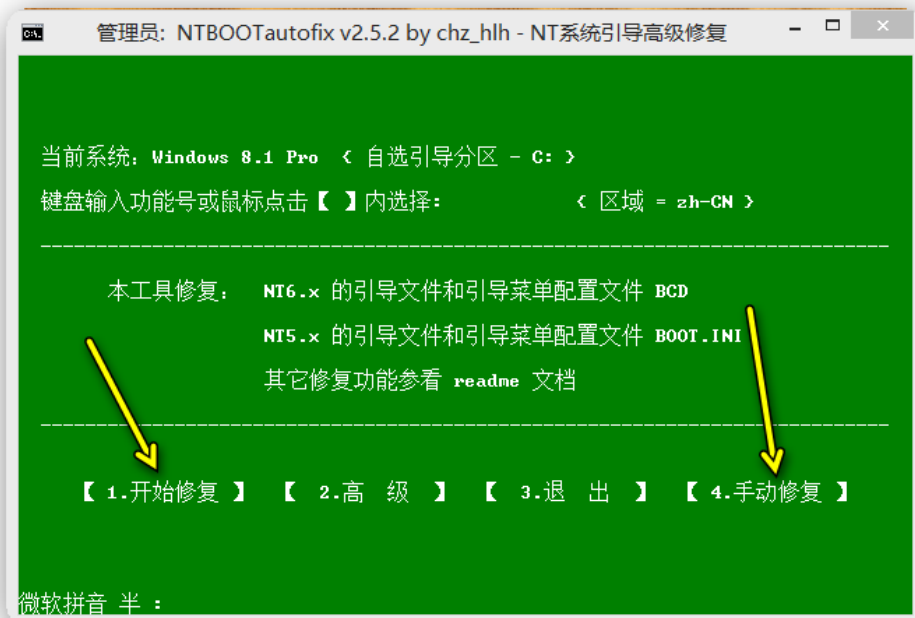
选择你的系统盘符，如下图



可以选择自动修复或手动修复

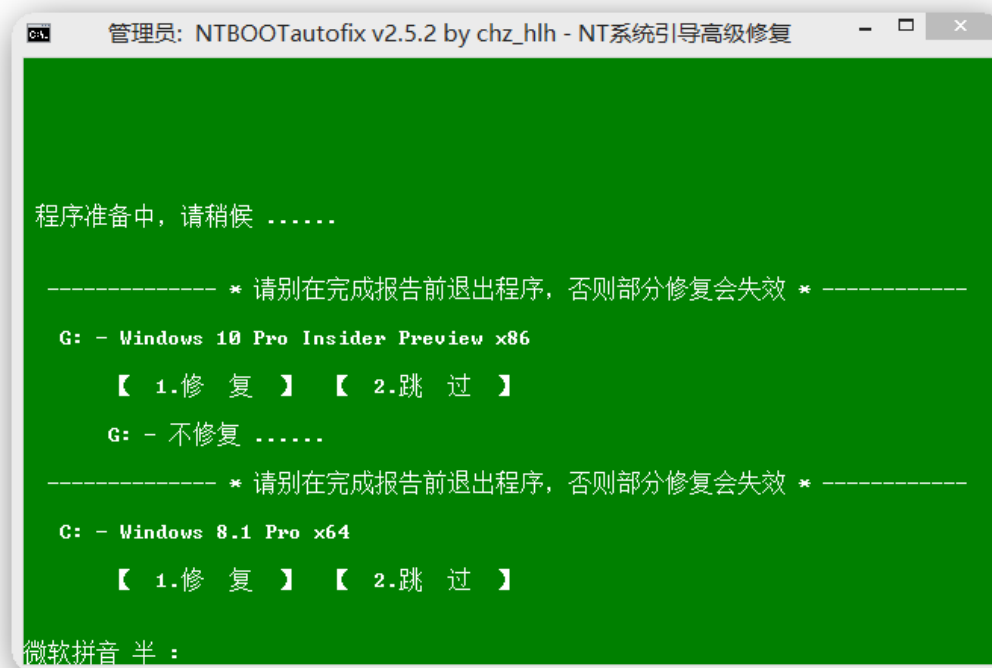
自动修复：会自动检测你的所有分区，然后查找到你的分区上已安装的系统自动进行修复

手动修复：检测你所有分区，查找到你的分区上已安装的系统，让你手动决定是否进行修复



如下图，我选择了手动修复，检测到我的 G 盘安装了 win10，提示我修复还是跳过，因为我的引导正常，为了给大家演示，我选择了跳过——不修复，然后检查到我的 C 盘安装了 win8.1





必须等到修复报告建立完成，才可以点击“退出”完成引导修复

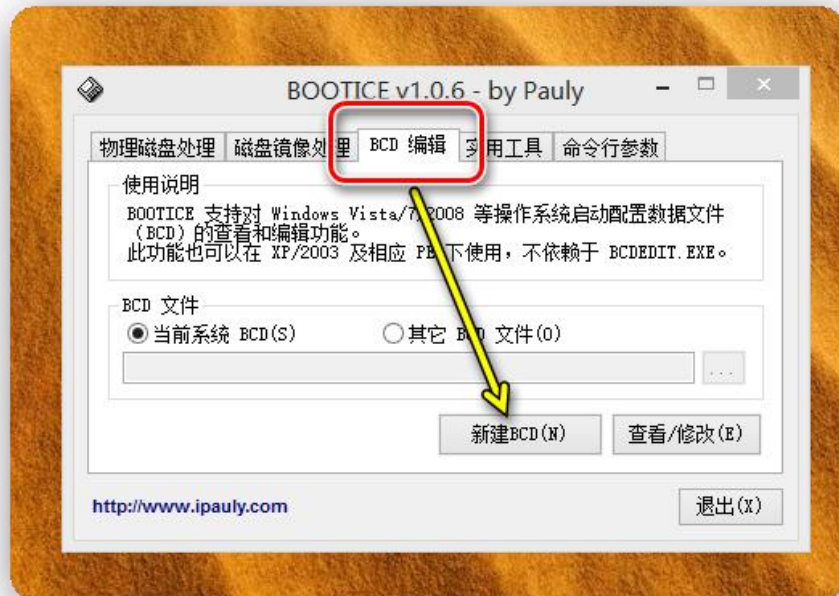


### 【注意】

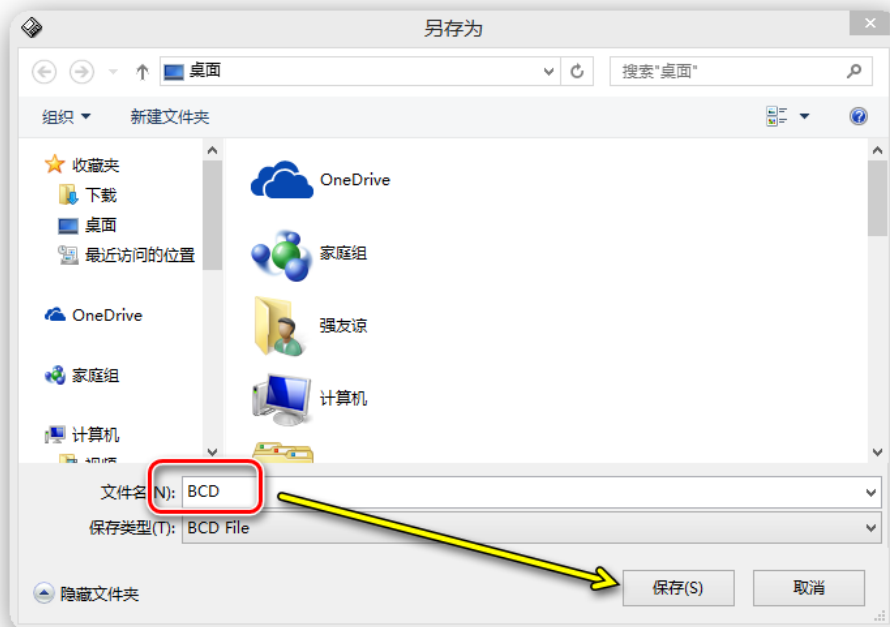
使用该软件进行 UEFI+GPT 模式系统引导修复时，需要先挂载 ESP/EFI 分区，可通过 DG 等工具为 ESP/EFI 分区指定盘符挂载，打开 NTBOOT 引导修复工具，在里面选择 ESP/EFI 分区所在盘符，修复即可

 方法二：使用 BOOTICE 工具来修复引导

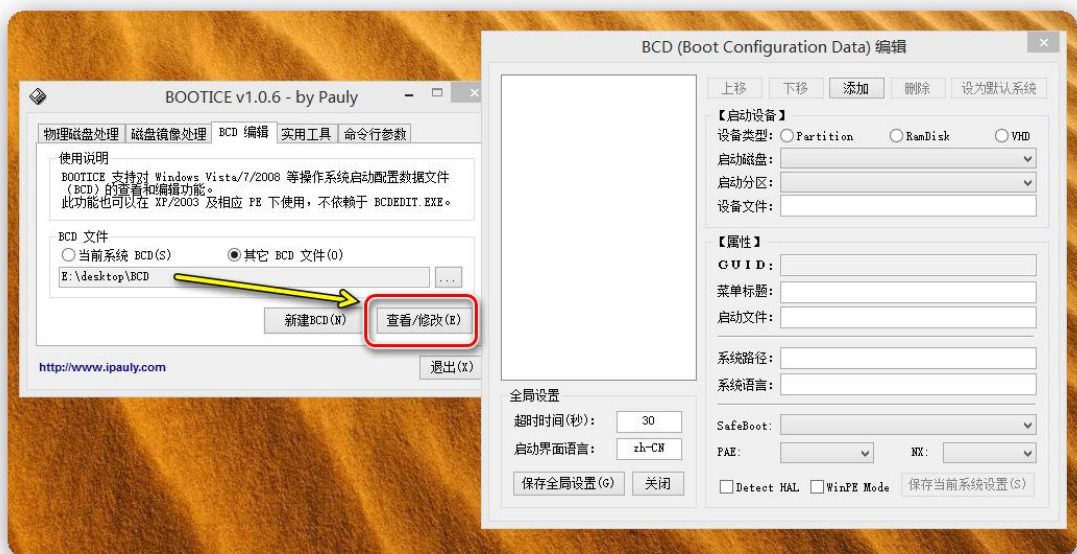
此工具不是专门用来修复引导的，其功能很是强大，这里只讲如何借助它修复引导  
这里我以修复 UEFI+GPT 模式下 win8 和 win10 双系统引导为例说明，Legacy+MBR 类似  
运行 BOOTICE 后，切换到“BCD 编辑”，然后“新建 BCD”



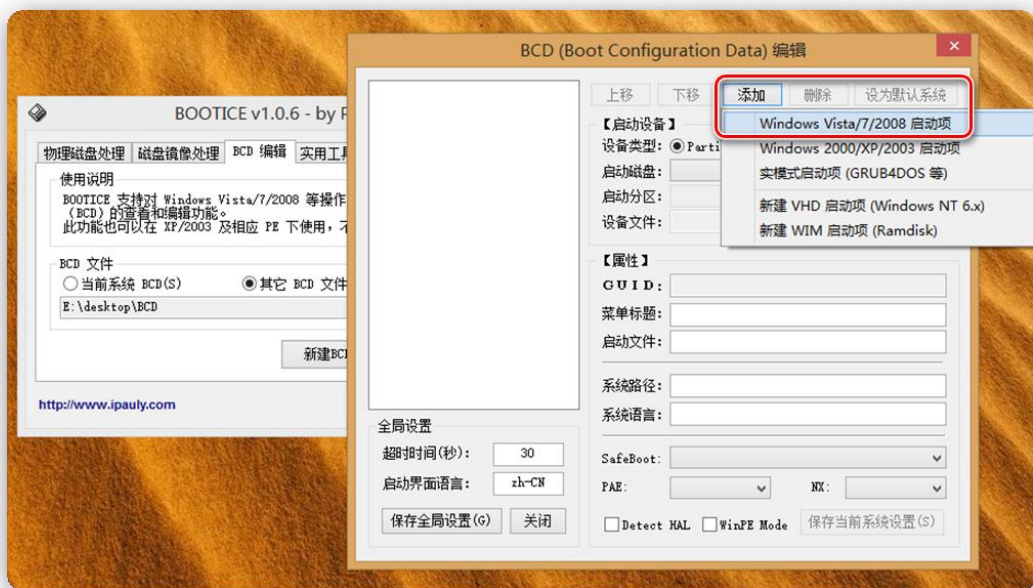
新建 BCD，文件名为:BCD



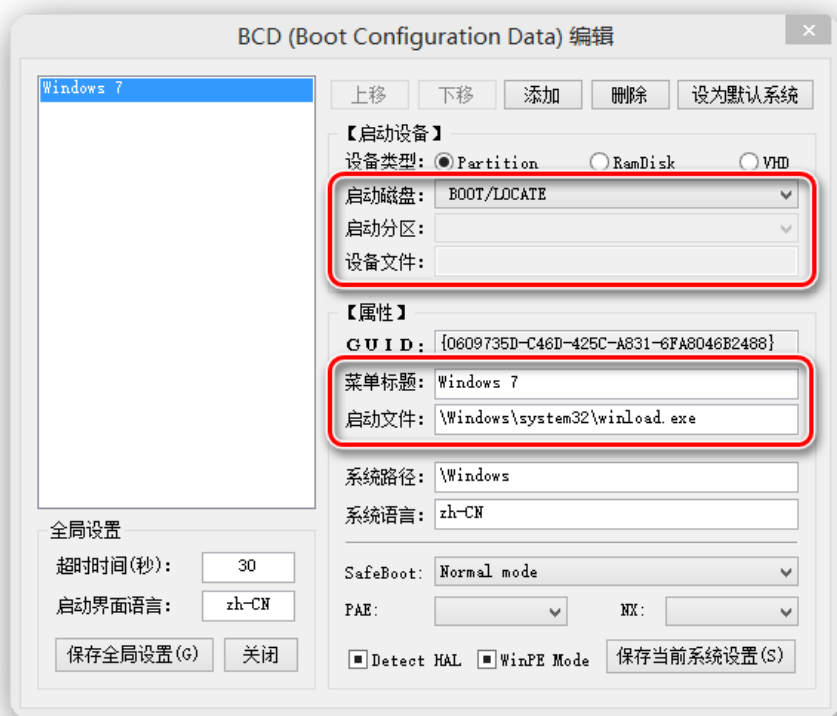
然后点击“查看/修改”



点击“添加”，选择“windows vista/7/2008”（这是 NT6.x 架构系统，当然 win8/10 也适用）

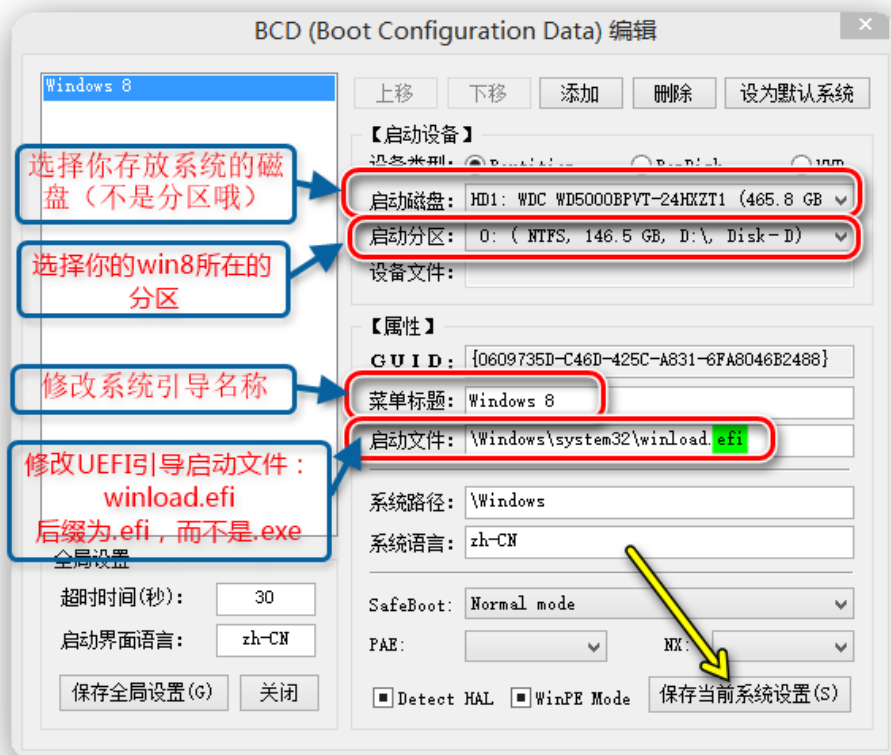


点击“添加”后，如下图为默认的初始 BCD 内容，在此基础上，我们进行 win8+win10 BCD 的修改

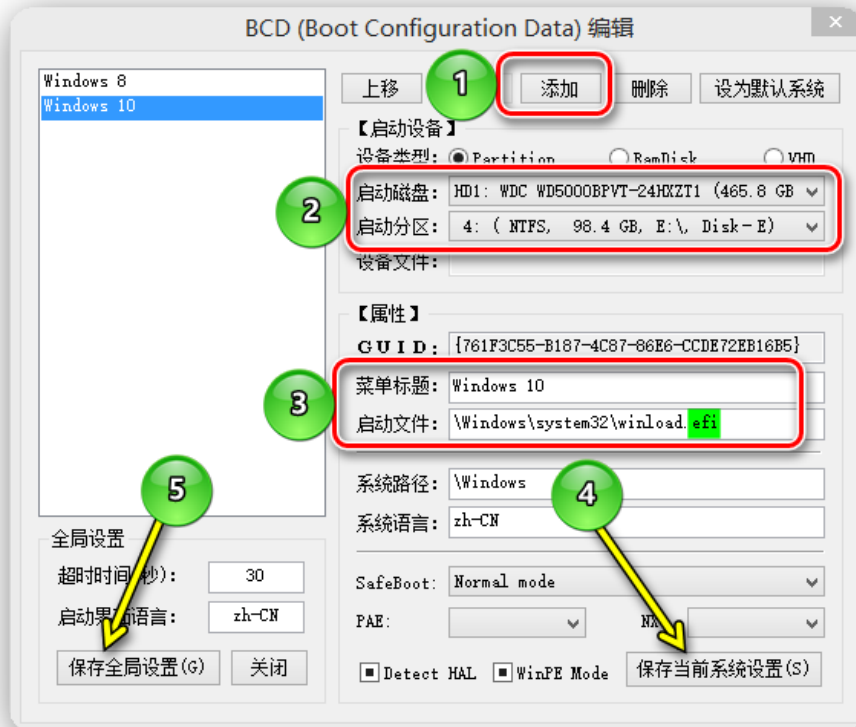


按照下图说明进行各项修改，启动文件 winload.efi 为 UEFI+GPT 启动文件，如果是修复 Legacy+MBR 启动文件，此处应为 winload.exe。

修改后，点击“保存当前系统设置”，这样，win8 的 BCD 内容就添加完成了



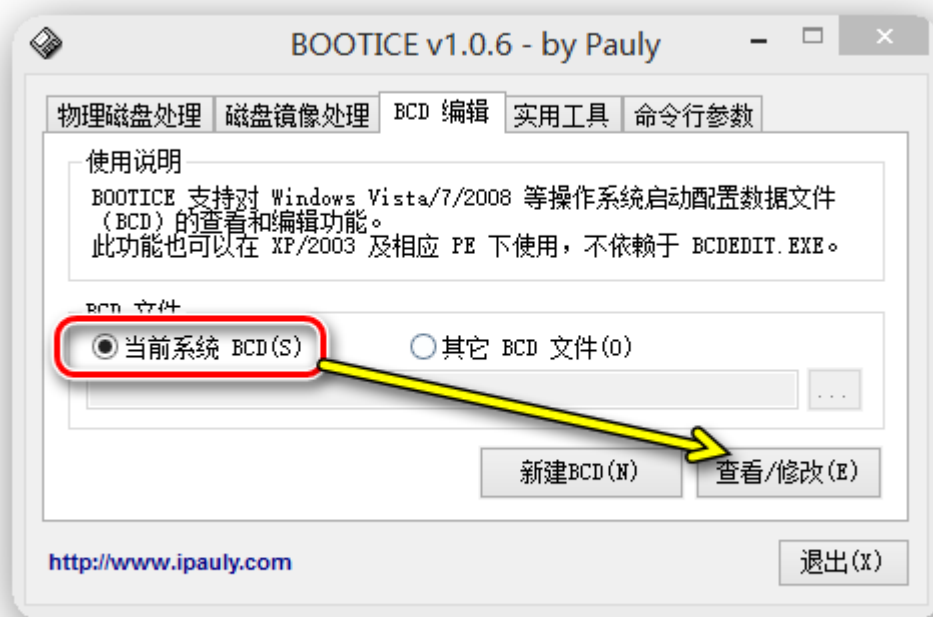
接下来，我们按照上面的方法再次添加 win10 的 BCD 信息，修改好之后，保存当前系统设置，win10 的 BCD 信息就添加好了；然后再点击“保存全局设置”，这样，win8 和 win10 的引导信息就会保存到我們创建的这个 BCD 文件中了，如下图：



创建好 BCD 文件后,我们只需在 PE 下将这个 BCD 替换到相应的目录下就可以完成引导修复了。上面我们是以 UEFI+GPT 模式举例的,所以此处 BCD 替换文件的目录为 ESP/EFI 这个 fat 分区: `efi\Microsoft\BCD`

如果是 Legacy+MBR 模式，若磁盘有一个隐藏的活动主分区，我们需要先给这个隐藏的主分区添加盘符（PE 下磁盘管理添加盘符或借助 DG 工具添加），然后将该 BCD 文件替换到这个活动主分区：**Boot\BCD**

方法二，同样适用于单系统 BCD 文件的修改；也可以不新建 BCD 文件，直接对系统中的 BCD 文件进行修改，就省去了替换 BCD 的步骤，如下：



之后的步骤跟上面一样。

NABOOT Autofix、BOOTICE、DG 等工具一般 PE 下都会集成的，这里就不给大家放单独的下载链接了。PE 制作及使用的话，这里就不说了，记得 N 久 N 久 N 久以前就发过专门的教程了，如果你非要问我关于 PE 的东西，请百度“电脑店”“大白菜”“老毛桃”等关键字。

最后说明下，方法二看起来要比方法一复杂了许多，但是我为什么还要费如此功夫介绍呢？因为，通过方法二，大家会更加清楚理解 UEFI+GPT、Legacy+MBR 引导的运行机制，这样就可以以不变应万变了！

另外，如果你是系统折腾狂的话，通过添加修改 BCD 文件，你甚至可以制作出支持 UEFI 和 Legacy BIOS 的 win7+win8+win10 多合一启动 U 盘，从而实现一盘在手，天下我有——一个 U 盘就可以搞定多个系统的安装！