**启动配置数据（BCD）存储**

**启动配置数据（BCD）存储（1）**

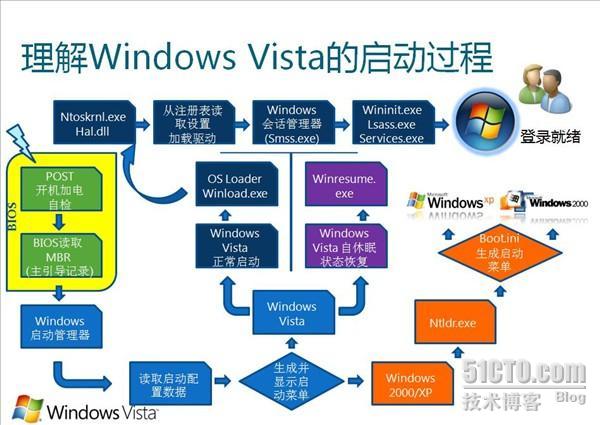
最近在看Windows 7和Windows Server 2008的部署，需要用到BCDEdit.exe创建Windows PE启动环境配置数据（BCD），在网上找了一些资料，大部分都是讲在多重启动环境中使用BCDEdit.exe编辑启动菜单的，但对于大多数应用不需要多重启动环境http://iwantfly.blog.51cto.com/editor/icons/etc_05.gif，所以仔细研究了一下BCD和BCDEdit.exe。下面是学习总结：

Microsoft从Windows Vista开始完全重新设计了操作系统启动环境，以解决现在硬件和固件日益增长的复杂性和多样性。重新设计 的操作系统启动环境的一个方面是新的独立于固件的数据存储，数据存储包含影响启动过程的配置数据。

要理解什么是启动配置数据（Boot Configuration Data – BCD），首先需要了解Windows启动过程。

*因为配置数据存储出现于Windows Vista，应用于Windows Vista及以后版本的Windows，所以本文中的Windows指的是Windows Vista及以后版本的Windows。另外，本文主要讨论基于BIOS系统计算机的启动配置数据（BCD）存储。*

**Windows启动过程**

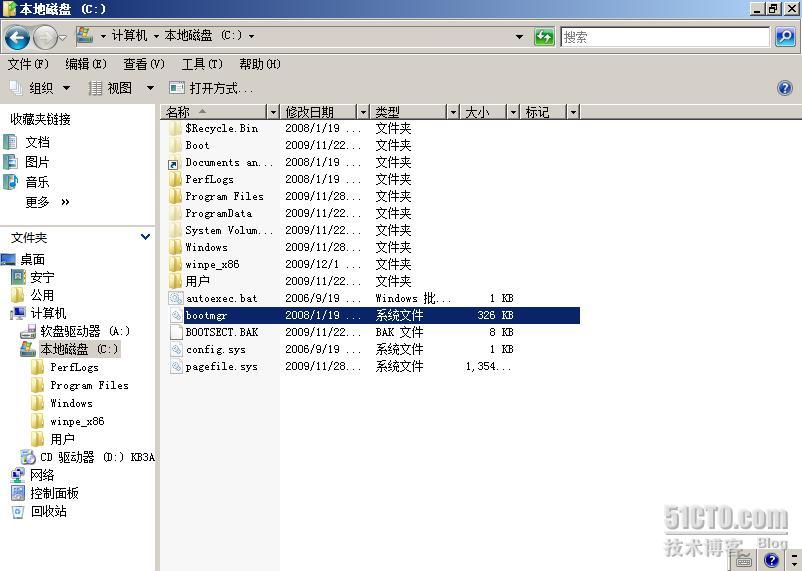
****

1、开启计算机电源，计算机加电自检（Power On Self Test – POST）

2、BIOS读取主引导记录（Master Boot Record – MBR），MBR读取引导扇区（boot sector）。引导扇区包含用以启动Windows启动管理器（Windows Boot Manager）程序的代码

3、Windows启动管理器读取系统启动配置数据（BCD）存储中的信息，系统启动配置数据存储包含存储启动时需要的配置信息。

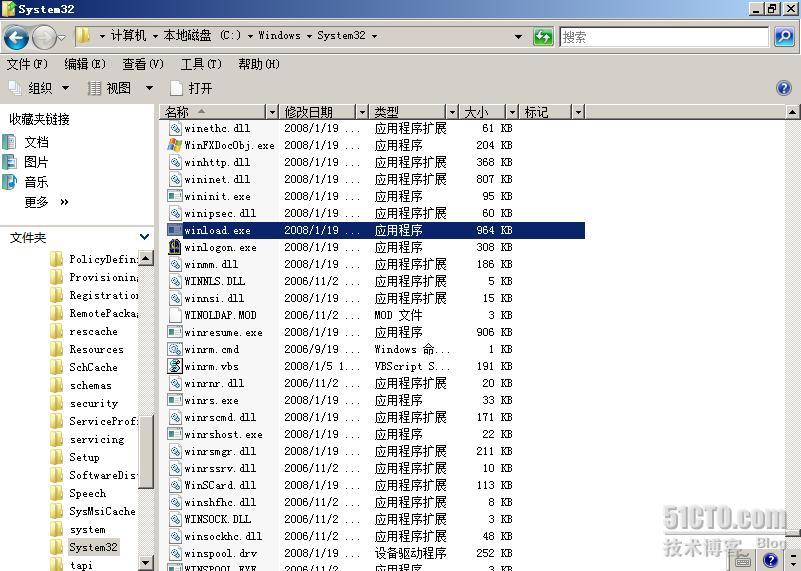
Windows启动管理器程序文件名为bootmgr，无扩展名，位于活动分区根目录



3、此时有两种情况：

（1）如果计算机只安装了一个版本的操作系统，或timeout设置为0，则不显示启动菜单，Windows启动管理器调用Windows启动加载器（Windows Boot Loader）

Windows启动加载器位于%systemroot%\system32文件夹，文件名为winload.exe



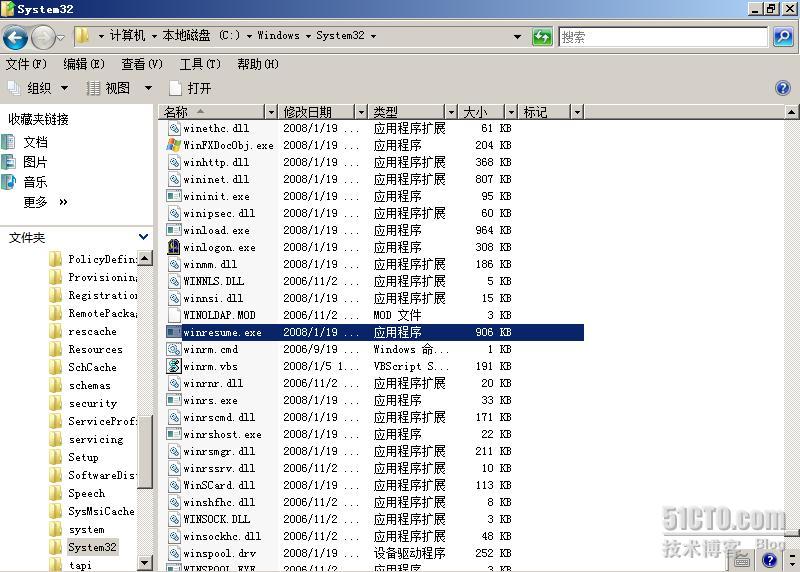
（2）如果计算机安装了多个版本的操作系统，则Windows启动管理器根据系统启动配置数据存储中的信息生成并显示启动菜单，以供用户选择

4、当用户在启动菜单中选择时：

Windows – Windows启动管理器运行Windows启动加载器

从休眠状态恢复 – Windows启动管理器运行Windows恢复加载器（Windows Resume Loader）从休眠恢复Windows运行状态

Windows恢复加载器位于%systemroot%\System32文件夹，文件名为winresume.exe



以前版本的Windows – Windows启动管理器定位所选系统的启动分区，并加载ntldr.exe，生成一个由boot.ini内容决定的启动菜单

5、Windows启动时，Windows启动加载器（winload.exe）加载ntoskrnl.exe和hal.dll，从注册表读取设置并加载驱动程序。接下来运行Windows会话管理（smss.exe），并且启动Windows启动初始化程序（wininit.exe）、本地安全验证（lsass.exe）与服务（services.exe）进程，完成后，用户就可以登录系统了。

从上述Windows启动过程可以看出，系统启动配置数据主要由Windows启动管理器在启动过程中使用，控制启动方式。

**术语**

**1、启动环境应用程序**

启动环境应用程序由BCD存储中的BCD对象（启动项）表示。启动环境应用程序的例子包括Windows启动管理器、Windows启动加载器、Windows恢复加载器、NTLDR、内存诊断程序……

**2、Windows启动管理器（Windows Boot Manager） - bootmgr**

一个启动环境应用程序，初始化启动过程。在一个多重启动环境中，启动管理器显示操作系统选择菜单，使用启动数据配置定位Windows启动加载器（winload.exe），Windows启动加载器加载特定版本的Windows。

**3、Windows启动加载器（Windows Boot Loader） - winload.exe**

一个启动环境应用程序，由Windows启动管理器调用，以加载操作系统内核ntoskrnl.exe及启动类（boot-class）设备驱动程序

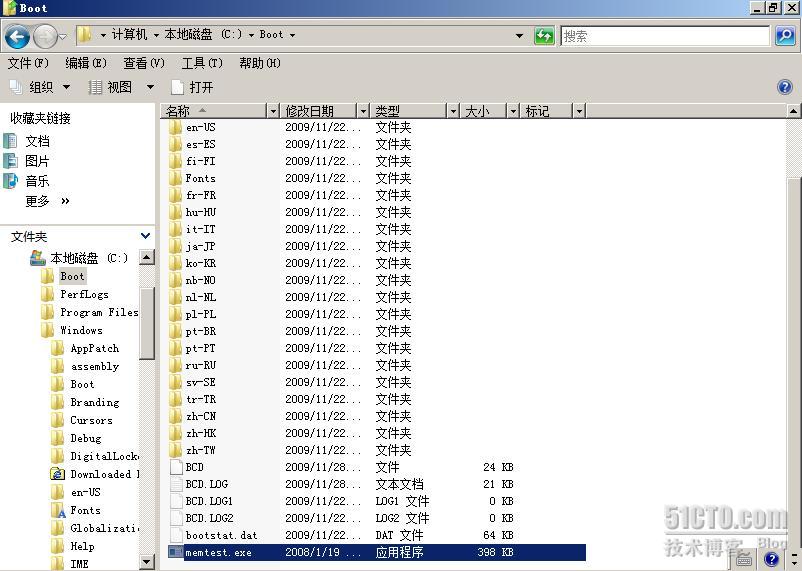
**4、Windows恢复加载器（Windows Resume Loader） - winresume.exe**

一个启动环境应用程序，将Windows从休眠状态恢复到运行状态

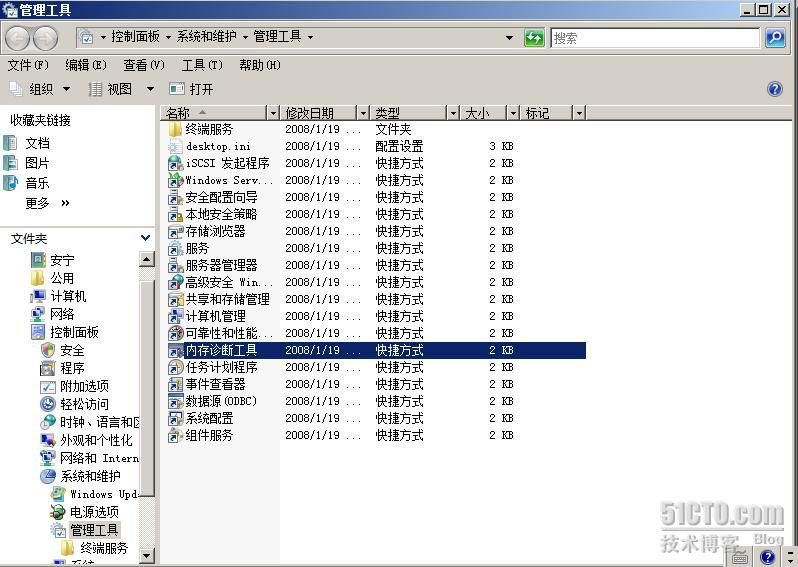
**5、内存诊断应用程序（memtest.exe）**

一个启动环境应用程序，运行内存诊断测试。通常称为memdiag

位于%systemdrive%\boot文件夹



在%systemroot%\system32文件夹中还有一个mdsched.exe文件，对应控制面板中的内存诊断工具



Mdsched.ee虽然也能够运行内存诊断，但从文件名就可以看出来mdsched.exe实际上是一个计划任务，用于向BCD存储中添加一个{memdiag}启动项，在计算机下次启动时实际运行的Windows内存诊断程序memtest.exe。

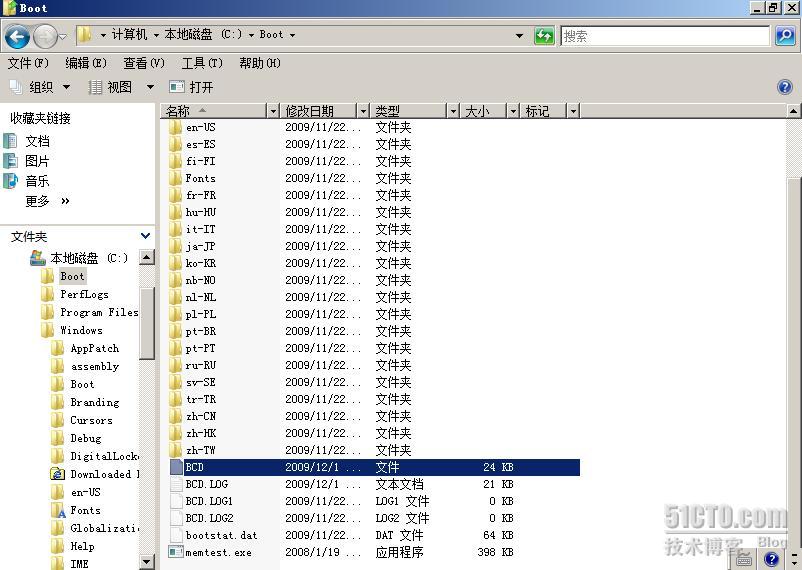
所以实际的内存诊断程序是memtest.exe

**6、NTLDR**

以前版本Windows的启动加载器，加载以前版本的Windows

**7、系统启动配置数据（BCD）存储**

一个独立于固件的数据库。系统启动配置数据保存在一个数据文件中（本质上是一个注册表配置单元文件），文件名为bcd，位于%systemdrive%\boot文件夹。启动配置数据（BCD）存储描述启动环境应用程序和启动环境应用程序的设置。启动环境应用程序在启动过程中要使用系统BCD存储中的信息。能够创建额外的BCD存储，但系统BCD存储只有一个



**8、启动项**

BCD存储中的BCD对象。BCD存储中包含多种类型的启动项。启动项还可以用于其他目的，如全局内存错误（RAM defect）列表或全局启动加载器（boot loader）设置

**9、数据类型/元素**

每个启动项具有的一个或多个数据值。因为每个元素有两个特性：名称和关联的数据类型，所以元素也称为数据类型，例如启动管理器有一个timeout元素（数据类型）

**10、标识符**

与每个启动项关联的全局唯一标识符（Global Unique Identifier – GUID）。

**11、well-known标识符**

常用启动项的标识符，作为GUID的别名。例如boot manager的well-known标识符为{bootmgr}；还有一些虚拟标识符，虚拟标识符的标识在每次启动时是可变的，例如{current}是当前启动的操作系统的well-knonw标识符。

    下一篇详细介绍启动配置数据以及如何使用BCDEdit.exe修改启动配置数据。

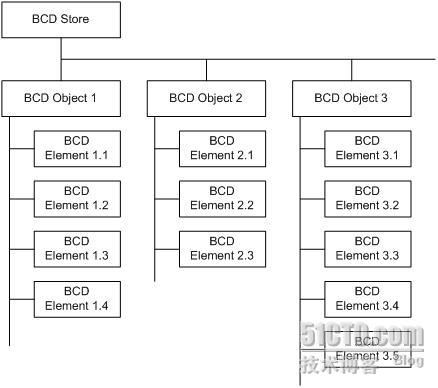
**启动配置数据（BCD）存储（2）**

**启动配置数据（Boot Configuration Data – BCD）**

启动配置数据为控制启动环境数据提供了一个独立于固件的机制。Windows Vista及以后版本的Windows都使用BCD加载操作系统或运行启动环境应用程序（如memtest.exe – 内存诊断）。BCD影响系统启动的所有方面。

BCD的数据存储是一个注册表配置单元格式的二进制文件，但不能使用注册表API访问。

**BCD体系结构**



BCD的体系结构是一个由三个基本组件构成的层次结构：存储、对象、元素。

BCD存储是层次结构中的顶层组件，BCD存储是BCD对象和元素的命名空间容器（namespace container）。

BCD对象是BCD元素的容器。常见BCD对象的类型描述启动环境应用程序，如Windows启动加载器（boot loader）的一个实例

BCD元素是一个数据项，如一个调试程序设置、一个启动环境应用程序名称或一个操作系统设置。

**BCD存储**

BCD存储是保存加载Windows或运行其他启动环境应用程序所需信息的BCD对象和元素的命名空间容器（namespace container）。计算机有一个描述所有安装的Windows操作系统和安装的Windows启动环境应用程序的系统BCD存储，除了系统BCD存储之外，还可以有多个非系统BCD存储

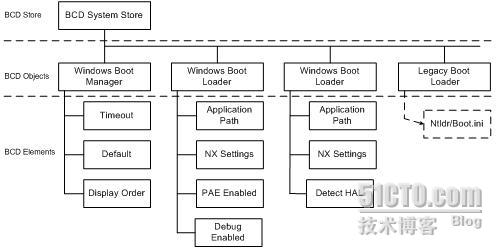
系统BCD存储文件夹名为bcd，位于活动分区的\boot文件夹

Windows启动管理器（Windows Boot Manager – bootmgr）使用系统BCD存储来控制启动过程。在多重启动环境中，Windows启动管理器使用来自系统BCD存储的信息显示启动选择菜单

BCD有两个接口：BCD WMI提供程序和BCDEdit.exe

管理员、技术支持人员能够使用BCDEdit.exe或BCD WMI创建额外的BCD存储，额外的BCD存储对于恢复、修复、捕获映像很有帮助，而且可以将非系统BCD存储导入为系统BCD存储

下图显示了一个典型BCD存储中BCD的层次



系统BCD存储中至少包含两个BCD对象：

1、Windows boot manager对象：Windows boot manager对象包含与Windows启动管理器相关的BCD元素，这些BCD元素包括显示在操作系统选择菜单、启动工具选择菜单中的启动项、选择菜单显示时间……。在使用BCDEdit修改一个BCD存储时，可以使用别名{bootmgr}

2、Windows boot loader对象：对于安装到计算机上的每个版本或配置的Windows，在BCD存储中都有一个Windows boot loader对象的实例。Windows boot loader对象包含加载Windows或Windows初始化时使用的BCD元素，这些BCD元素包括no-execute（NX）页面保护策略、物理地址扩展（Physical Address Extension – PAE）策略、内核调试程序设置……。当计算机启动到Windows时，新联的Windows boot loader对象实例由别名{current}标识。当使用BCDEdit修改BCD存储时，默认的Windows boot loader对象由别名{default}标识

系统BCD存储中也可能包含下列BCD对象：

Windows NTLDR对象：NTLDR对象描述以前版本Windows启动加载器（Ntldr.exe）的位置，Ntldr.exe用于启动以前版本的基于NT内核的Windows。只有计算机中安装了以前版本的Windows时才需要NTLDR对象。NTLDR对象的别名为{ntldr}

启动环境应用程序：BCD存储能够包含执行其他与启动相关操作的BCD对象，如Windows Memory Tester（进行内存诊断）

**BCD对象**

有三类BCD对象：application对象、inheritable对象、device对象。最常见的对象类型是application对象，application对象描述启动环境应用程序，如Windows启动管理器、Windows启动加载器。每个BCD对象由一个128位的全局唯一标识符（GUID）标识。

**Application对象**

一个application对象代表一个启动环境应用程序。标准application对象包括：

Windows boot manager对象 – {bootmgr}

Windows boot loader对象 – {current} {default}

Windows NTLDR对象 – {ntldr}

Windows resume loader对象

Windows Memory Tester对象 – {memdiag}

**Inheritable对象**

一些BCD元素能够应用于多个application元素，有些BCD元素对BCD存储是全局性的。虽然能够将这些元素分别与使用它们的每个对象实例分别关联，但更有效的方法是创建一个inheritable对象，作为在多个BCD对象实例之间共享的元素的容器。其他对象在使用这些BCD元素时，只需要引用inheritable对象，而不是每个元素，即inheritable对象包含能够应用于多个对象的元素。

Inheritable对象只能包含应用于所有启动环境应用程序的BCD元素。

下面是一些常用的inheritable对象：

全局RAM错误列表，能够被所有启动环境应用程序继承 – {badmemory}

启动加载器全局设置，应当被所有Windows boot loader应用程序继承 – {bootloadersettings}

全局调试程序设置，能够被所有启动环境应用程序继承 – {dgbsettings}

全局紧急管理服务（Emergency Management Services – EMS）设置，能够被所有启动环境应用程序继承 – {emssettings}

全局设置，应当被所有启动环境应用程序继承 – {globalsettings}

恢复加载器全局设置，应当被所有恢复应用程序继承 – {resumeloadersettings}

**Device对象**

Device对象描述没有存储在一个application对象中的设备特定的属性。

大多数设备，如硬盘分区，能够由一个BCD元素描述，但复杂的设备需要多个BCD元素来描述。

例如，当从一个Windows映像（WIM）文件创建的RAM磁盘启动时，就需要device对象，此device对象包含WIM文件的位置，如果是从网络启动的，此device对象还包含网络端口信息。RAM磁盘对象的别名为{ramdiskoptions}

可以创建自定义的device对象。

**BCD元素**

BCD元素是启动环境应用程序或Windows启动过程中的配置数据。

一些BCD元素只能与特定BCD对象关联，有些BCD元素能够应用于所有类型的启动环境应用程序。

**总结**

BCD存储由一系列对象组件，每个对象是一个启动环境应用程序，每个对象对应一个启动项。每个对象由一系列元素组成，每个元素有一个属性。

简单地说，可以将BCD元素理解成BCD对象的属性。

**管理BCD的工具**

BCDEdit.exe

BCD WMI API

控制面板 - 系统，允许用户指定两个全局BCD元素的值：默认操作系统和配置、启动管理器超时设置

Msconfig.exe

**BCDEdit.exe**

BCDEdit.exe（位于%systemroot%\system32文件夹）是一个命令行工具，用于管理BCD存储。BCDEdit.exe能够新建、修改BCD存储、添加启动菜单选项……

使用BCDEdit.exe修改BCD存储需要管理员权限。

BCDEdit.exe命令划分为8类：

**1、对BCD存储进行操作的命令**

/createstore：新建空BCD存储

/export：将系统BCD存储的内容导出到文件

/import:使用由/export导出的文件还原系统BCD存储的状态

**2、对存储中的启动项进行操作的命令**

/copy：复制BCD存储中的启动项

/create：在BCD存储中新建启动项

/delete：删除BCD存储中的启动项

**3、对元素进行操作的命令**

/deletevalue：删除启动项的元素

/set：创建或修改一个启动项的元素（这是我要详细说明的命令，因为使用Windows PE时，这是一个最主要的命令）

**4、控制输出的命令**

/enum：列出一个BCD存储中的启动项

/v：完整显示启动项的GUID，而不是well-known标识符

**5、控制启动管理器的命令**

/bootsequence：指定一次性启动顺序，例如在使用控制面板中的内存诊断工具（mdsched.exe）时，就是通过指定一次性启动顺序在计算机重启时，运行memtest.exe命令来进行内存诊断的

/default：指定默认启动项

/displayorder：指定启动管理器显示启动菜单时启动项的顺序

/timeout：指定启动管理器超时值

/tooldisplayorder：指定启动管理器显示工具菜单时启动项的顺序

**6、控制紧急管理服务（EMS）的命令**

/bootems：为一个特定启动应用程序启用或禁用EMS

/ems：为一个操作系统启动项启用或禁用EMS

/emssettings：设置全局EMS参数

**7、控制调试的命令**

/bootdebug：为一个启动应用程序启用或禁用调试

/dgbsettings：设置调试程序的全局参数

/debug：为一个操作系统启动项启用或禁用内核调试

**8、修改其他命令的命令**

/store：指定命令作用到的BCD存储。如果不指定，默认是作用于系统BCD存储。

**实例**

注意：对系统BCD存储进行修改可能会影响系统的正常启动，所以尽量在虚拟机上进行实验，或在修改系统BCD存储之前先备份系统BCD存储，这样在系统不能正常启动时，可以使用备份的系统BCD存储进行恢复。

**一、BCD存储内容**

查看指定BCD存储内容的命令

Bcdedit [/store *文件名*] /enum [*类型* | *id*] [/v]

/store *文件名*：可选。指定BCD存储。如果不指定，默认系统BCD存储

*类型*：可选。指定要查看的启动项的类型。

active – 启动管理器启动菜单中的所有启动项

all – 所有启动项

bootapp – *所有启动环境应用程序*

inherit – 所有inheritable对象

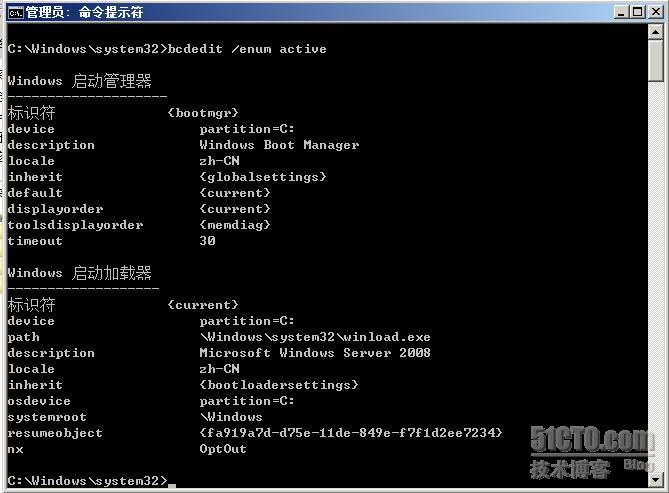
osloader – 所有Windows boot loader启动项

resume – 所有休眠恢复启动项

*Id*：可选。要查看的启动项的标识符，可以使用GUID或well-knonw标识符

/v：可选。在输出时显示GUID，而不是well-known标识符

因为系统BCD存储、/enum、active是默认选项，所以如果只输入bcdedit，实际上对系统BCD存储运行bcdedit /enum active



从前面的输出可以看出：

启动环境被划分为两个类型：Windows启动管理器（Windows Boot Manager）和在启动环境中运行的其他启动应用程序，统称为启动项。Windows Boot Manager基本上是一个mini操作系统，控制启动，并允许选择要运行的启动环境应用程序。有许多启动环境应用程序（例如Windows启动加载器 – Windows boot loader……），每个启动环境应用程序的作用不同。

详细解释一下bcdedit /enum active的输出

|  |  |  |
| --- | --- | --- |
| **Windows启动管理器** | | |
| 标识符 | {bootmgr} | Windows boot manager对象的标识符 |
| device | partition=c: | Windows启动管理器程序（bootmgr）位于c:盘 |
| description | Windows Boot Manager | Windows启动管理器对象的描述 |
| locale | zh-CN | Windows启动管理程序的区域设置 |
| inherit | {globalsettings} | Windows启动管理器对象应当继承{globalsettings}对象的设置 |
| default | {current} | Windows启动管理器的默认启动项的标识符，即当前启动操作系统的Windows boot loader |
| displayorder | {current} | Windows启动管理器显示启动菜单时启动项的顺序 |
| tooldisplayorder | {current} | Windows启动管理器显示工具菜单时启动项的顺序 |
| Timeout | 30 | Windows启动管理器等待时间，如果超时，Windows启动管理器运行default指定的启动项 |
| **Windows启动加载器** | | |
| 标识符 | {current} | Windows boot manager对象的标识符，{current}表示当前启动的操作系统的Windows boot loader |
| device | partition=c: | device元素和path元素结合说明Windows boot manager是C:\windows\system32\winload.exe |
| path | \windows\system32\winload.exe |
| description | Microsoft Windows Server 2008 | Windows boot manager对象的描述 |
| locale | zh-CN | Windows boot manager的区域设置 |
| inherit | {bootloadersettings} | Windows boot manager要继承{bootloadersettings}对象的设置 |
| osdevice | partition=c: | osdevice元素和systemroot元素结合说明操作系统文件位置 %systemroot%变量 |
| systemroot | \windows |
| resumeobject | {fa919a7d-d75e-11de-849e-f7f1d2ee7243} | 恢复应用程序的标识符 |
| nx | Optout | 对操作系统和所有进程启用DEP（数据执行保护），包括操作系统内核和驱动程序 |

osdevice和systemroot元素我觉得有些不好理解，思考了挺长时间，按我自己的理解解释一下，大家讨论

1、一台计算机上只有一个启动管理器，即BCD存储中只能有一个Windows boot manager对象，此对象包含全局信息：哪个启动项是默认启动项（默认启动哪个启动环境应用程序）、启动菜单显示时间、启动项在启动菜单中的显示顺序。

2、一台计算机上可能有多个启动加载器，对于计算机上安装的每个Windows，都有一个启动加载器。例如，如果计算机上安装了两个不同版本的Windows Vista，计算机上就有两个Windows启动加载器。所以BCD存储中可能有多个Windows boot loader对象

举例如下：

一台计算机上安装了Windows Vista Home Basic（操作系统文件安装在C:\windows）和Windows Vista Enterprise（操作系统文件安装在D:\windows）

计算机中就有了两个Windows boot loader（winloade.exe），一个位于c:\windows\system32，用于启动Windows Vista Home Basci；一个位于d:\windows\system32，用于启动Windows Vista Enterprise。

这样在BCD存储中就有了两个Windows boot loader对象，而且当在启动菜单中选择时，这两个boot loader要使用osdevice和systemroot元素找到自己对应的操作系统文件进行启动。

Osdevice和systemroot元素也是在自定义Windows PE时经常要遇到的两个元素。

**启动配置数据（BCD）存储（3）**

**二、bcdedit /set命令**

编辑BCD存储时最常用的命令是bcdedit /set，下面用两个实例说明bcdedit /set命令

Bcdedit [/store *文件名*] /set [*id*] *数据类型 值*

为一个指定的启动项新建一个元素或修改一个已有的元素。

/store *文件名*：可选。指定BCD存储。默认系统BCD存储

*id*：可选。要修改的启动项的标识符。默认{current}

*数据类型*：必需。要创建或修改的BCD元素。不同的启动项有不同的BCD元素

*值*：必需。BCD元素的值。值的格式取决于BCD元素。

例如：bcdedit /set nx optin

此命令将当前Window boot loader对象的nx元素设置为optin，即将操作系统的DEP（数据执行保护）设置为只为操作系统组件（包括操作系统内核和驱动程序）启用DEP

Bcdedit /set命令涉及到的数据类型很多，使用bcdedit /? types *apptype*命令查看指定启动项关联的BCD元素

*apptype*指定显示启动项

常见的数据类型：

bootapp：所有启动应用程序

bootmgr：启动管理器

devobject：device对象

memdiag：内存诊断应用程序

osloader：Windows启动加载器（{current}、{default}）

resume：休眠恢复应用程序

customtypes：自定义数据类型

ntldr：NTLDR

bootsector：启动扇区应用程序，启动扇区应用程序允许Windows启动管理器启动一个与非Windows操作系统关联的16位实模式应用程序

每个数据类型（BCD元素）的值有不同的格式，使用bcdedit /? formats命令查看

布尔：可能的取值为true（1、on、yes）或false（0、off、no）

设备：可能的取值为boot、partition=<*drive*>、file=[<*parent*>]<*path*>、ramdisk=[<*parent*>]<*path*>,<*optionsid*>

id：启动项标识符

整数：64位整数

整数列表：以空格分隔的一个或多个64位整数的列表

列表：启动项标识符列表

字符串：文本字符串

**下面使用《Windows PE用户指南》中的一个例子详细解释bcdedit.exe**

1、操作实例：从CD-ROM启动Windows PE

详细介绍bcdedit命令的用法，其他步骤简要列出

步骤1：设置Windows PE构建环境

步骤2：添加启动支持

步骤3：可选。添加其他自定义

**步骤4：配置BCD存储**

del c:\winpe\_x86\mount\boot\bcd

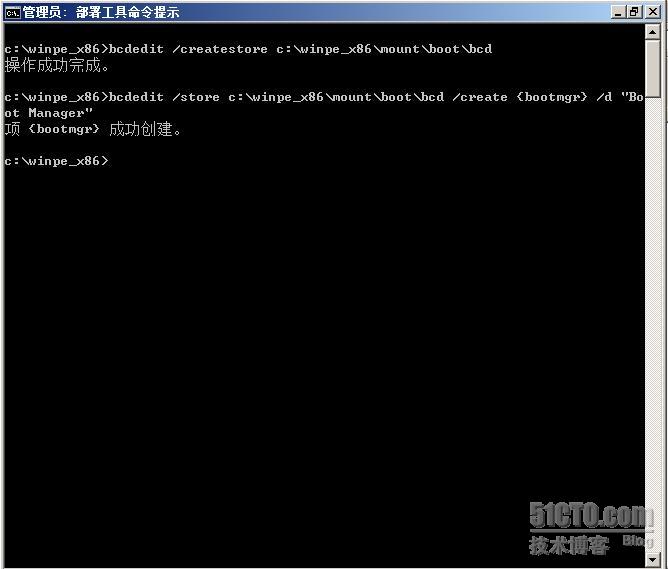
删除装载到c:\winpe\_x86\mount文件夹中的Windows PE映像的启动配置数据存储（\boot\bcd），以自定义Windows PE映像的启动配置数据

bcdedit /createstore c:\winpe\_x86\mount\boot\bcd

为装载到c:\winpe\_x86\mount文件夹中的Windows PE映像新建一个空的启动配置数据存储文件

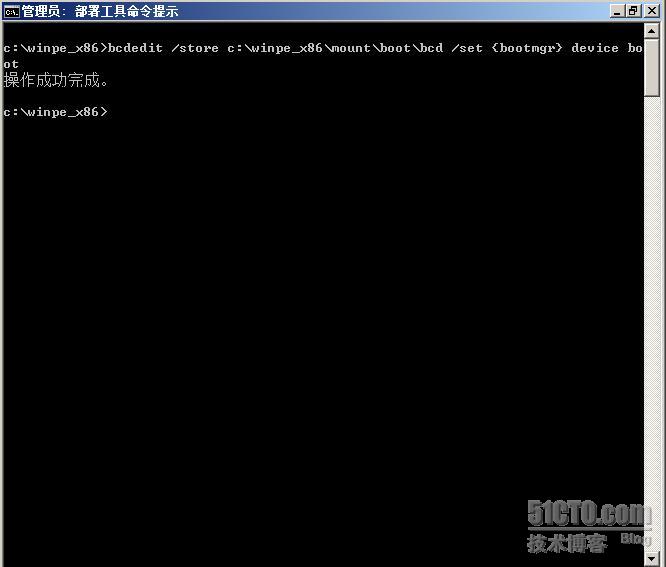
bcdedit /store c:\winpe\_x86\mount\boot\bcd /create {bootmgr} /d “Boot Manager”

在新建的空白启动配置数据存储中新建一个BCD对象（启动项 – Windows启动管理器）。必须指定要使用的BCD存储，如果不指定，修改的技术人员计算机的系统BCD存储



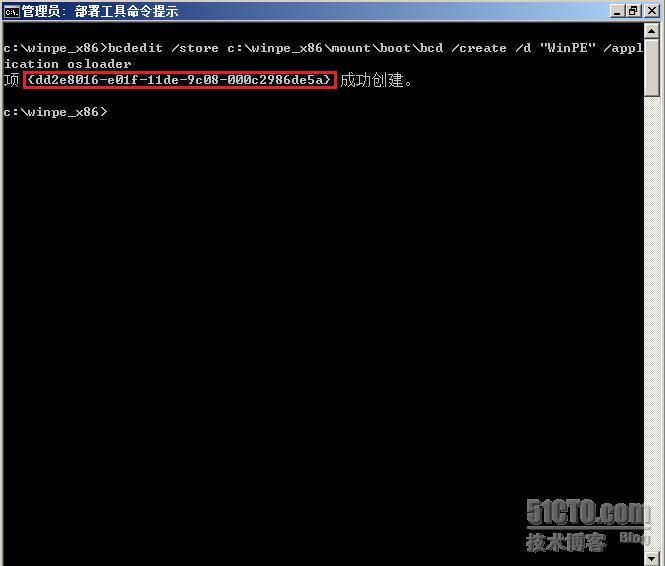
bcdedit /store c:\winpe\_x86\mount\boot\bcd /set {bootmgr} device boot

指定Windows PE的Windows启动管理器（bootmgr）位于启动设备（在此例中为光盘驱动器）



bcdedit /store c:\winpe\_x86\mount\boot\bcd /create /d “WINPE”/application osloader

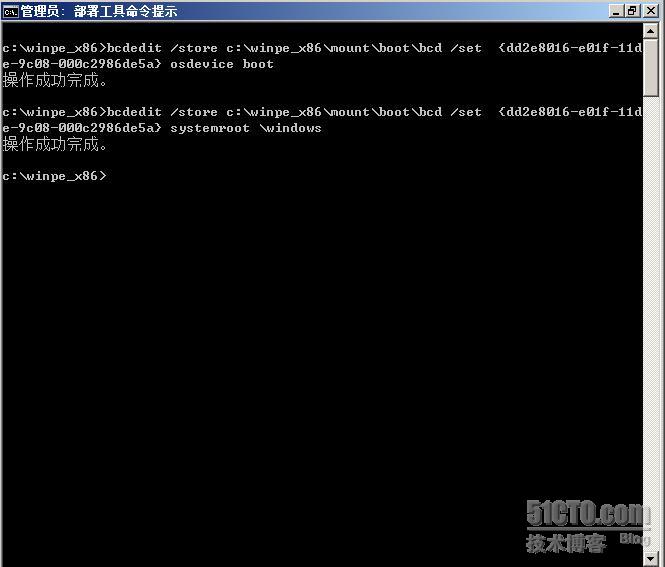
在Windows PE的启动配置数据存储中新建一个application对象（启动项 – Windows启动加载器）。由于没有指定新建的启动项的标识符，bcdedit新建一个GUID作为此启动项的标识符



bcdedit /store c:\winpe\_x86\mount\boot\bcd /set <GUID> osdevice boot

bcdedit /store c:\winpe\_x86\mount\boot\bcd /set <GUID> systemroot \windows

指定Windows PE操作系统文件位于启动设备（此例中为光盘驱动器）的windows文件夹



bcdedit /store c:\winpe\_x86\mount\boot\bcd /set <GUID> device boot

bcdedit /store c:\winpe\_x86\mount\boot\bcd /set <GUID> path \winodws\system32\winload.exe

指定Windows启动加载器是位于启动设备（此例中为光盘驱动器）的\windows\system32\winload.exe



bcdedit /store c:\winpe\_x86\mount\boot\bcd /set <GUID> winpe yes

计算机启动时能够启动到Windows PE

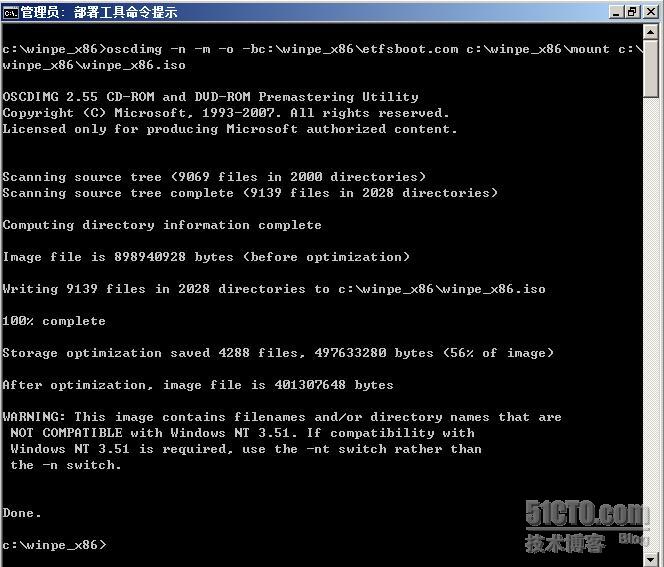


bcdedit /store c:\winpe\_x86\mount\boot\bcd /displayorder <GUID> /addlast

将Windows启动加载器添加到启动菜单最后



步骤5：创建一个可启动CD-ROM或DVD-ROM



进行测试



可以看出，Windows PE直接从光盘启动，而没有启动到Windows PE RAM磁盘，说明修改Windows PE的启动配置数据（BCD)成功！