**WIN7/XP用注册表关联指定后缀名和打开程序（手动【图文】和C编程两种实现）**

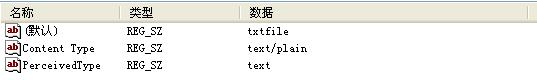
前言：

本文是基本原理介绍和手动的操作。程序实现该功能在<http://blog.csdn.net/arvon2012/article/details/7839556>，同时里面有完整代码的下载。

今天在开发程序的时候，需要做后缀和程序的关联，所以写下详细的操作过程跟大家分享。有什么意见欢迎留言！

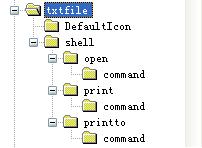
目标：我这次要自定义一个后缀名（.seve），并且自己开发一个负责打开这种后缀文件的程序（Opo.exe）。当双击后缀名为我自定义的文件时，自动运行我的程序（Opo.exe）。

**概要原理：**  
对windows来说，双击一个文件时，windows会根据他注册表中的记载打开和当前文件后缀名关联的程序。如默认情况下：.txt关联的是NODEPAD.exe，.doc关联的是word。所以要进行关联，就是要做注册表的修改。修改注册表，可以通过手动方式，也可通过注册表编程实现。

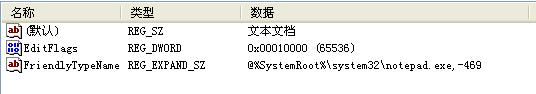
**那么注册表中是怎么把指定的后缀和指定的程序关联的？**  
我们用实例讲解，这里的系统是XP(WIN7系统的注册表和XP至少在这里用到的地方都是一样的)，点击开始——运行，输入regedit（这个命令用了很久，但是我比较笨，后来才看出来其实就是register edit（注册表编辑）的简写）。  
打开后可以看到几个“键”，我们关心的和后缀名相关的，他们在HKEY\_CLASSES\_ROOT下面。点开后，点编辑——查找，然后在查找栏中输入.txt就自动跳转到.txt所在的位置，大家在右边可以看到如下内容：  
  
按顺序讲下这三个值：  
1.（默认）：在关联中最重要的一个值，决定了当前后缀的文件交给谁并且怎么处理，这里大家看到的内容是“txtfile”。其实这个.txt键只是相当于一个索引，该键中并没有实际的处理说明，而实际的处理都交给了另一个键，那个键的名字就是这里的“（默认）”给出的。（着急的童鞋现在就可以在注册表中搜txtfile键）

2Content Type：这里指定了这个文件内容是什么类型，这个值可以接受很多参数，这个参数的列表可以再MSDN中查找Content Type看到。这个值存在的意义是当系统要处理这个文件内容的时候，和这个后缀名关联在一起的程序会根据这个值决定如何打开并展现文件中的内容。

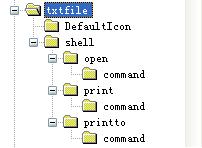
3.PerceivedType：这个值也是一个类型的标记。他的用处呢？举个例子大家就有感觉了，如果这个值的内容是video的时候。我们打开一些视频播放器，许多播放器第一次打开很可能会搜索电脑硬盘，寻找可以播放的文件并展现在“播放列表”中，而这个搜索过程就是根据这个值进行的。

下面跳转到负责真正处理和打开工作的键中，对于我们的例子.txt，这个处理键，上面已经说了，就是txtfile，找到该键，并且全部展开，可以看到如下内容：  
  
**PS**：在关联后缀名和打开程序中，都是两个键成对工作的，比如这里的.txt和txtfile键。我们暂且称第一个是“后缀键”第二个是“处理键”。这样方便后面表达。

上面的图中，大家可以看到txt文件的处理键的层次内容。选中最顶层的txtfile看右边界面显示的内容：



1.（默认）：这里是文件类型的说明，大家右击文件图标，选择属性的时候，在属性中的文件类型框中可以看到这个值。  
2.EditFlags：标识这个文件的读写权限，比如是只读，还是可读写。。。等等，这个在文件属性里也能修改。  
3.FriendlyTypeName：这个值是应用程序的友好名称（晕菜，什么是友好名称？这里先说下不友好名称是什么：应用程序的ID，没错，就是看也看不懂的那一串数字，所以你在这里看到的这个值已经很适合人脑去理解了，很友好！）。这个值其实就是说明这个处理键对应的应用程序是哪个，大家打开文件属性找一找，也能找到这个值的身影。不过记住，这个值可不是起决定作用。决定作用在子键中。

现在回到txtfile键层次图中:  
  
可以看到这个处理键的直接子键有两个：1.DefaultIcon这一类文件的默认图标。2.Shell，存放了对这类文件进行实际操作时，执行的命令。

我们直接看重点：

**shell键：**  
在这个键中，可以看到三个子键：open，print，printto，这里列出的可以执行的命令，其实和鼠标右击这个文件时候，弹出的菜单中的命令对应。我们关心的是打开操作，所以这里我们看下open键下的command里存的是什么：  
默认键：%SystemRoot%\system32\NOTEPAD.EXE %1  
看到木有？这里真正的指定了由谁来打开这种后缀的文件。这里是NOTEPAD.EXE。  
和open键平行的建都是一些操作，大家感兴趣可以一个个打开研究研究。

**实现关联**  
既然知道注册表中后缀和程序是怎么关联的，我们就也来关联一组。  
后缀：opo  
程序：Opo.exe

1.手动方式  
我们先手动创建，之后再说程序实现，原理上都一样，只不过程序的比较帅，当然也比较高端（在某些应用场合）。  
鼠标右击HKEY\_CLASSES\_ROOT，弹出菜单，选新建——项  
  
**1.后缀键**  
创建：然后会创建一个键，我们给它输入名字，这里的名字就是后缀名：.opo。  
写值：选中该键，然后在右边的界面中双击（默认），弹出对话框，我们在值中输入处理键的名字：opofile。  
后缀键完成！

2.**处理键**  
创建：和上面一样，名字写成上面的键的（默认）值opofile  
我们就是要关联打开应用，其他乱七八糟的东西我们先不关心，所以如下：  
  
右击opofile，在opofile键下创建键shell，shell下创建键open（定义打开命令），open下创建子键command。  
写值：选中command，然后双击右边的默认，在弹出框中的写入程序Opo.exe的位置后面加上%1，我的是这样的：



[**测试**](http://lib.csdn.net/base/softwaretest)**：**  
现在随便找个小程序，名字改成Opo.exe，然后自己创建一个文本文件，重命名成后缀为opo的文件。然后双击这个文件，额。。。。  
这就是所谓的后缀名和程序的关联，完成！嘿嘿

**WIN7/XP用注册表关联指定后缀名和打开程序(C程序实现)**

在前一篇文章中http://blog.csdn.net/arvon2012/article/details/7818909，已经介绍了用注册表关联指定的后缀名和打开程序的原理，而且了详细的手动操作过程。本文通过C程序实现上文的关联功能。

本文的完整代码在http://download.csdn.net/detail/arvon2012/4482694，

对刚接触注册表编程的朋友请看入门文章http://blog.csdn.net/arvon2012/article/details/7839659

这里其实就是用代码模拟手工操作，完全一样的步骤实现关联。有什么意见欢迎留言！谢谢！

步骤：

1.创建后缀键，并给后缀键传入默认值，设置好处理键

//create .seve

if (ERROR\_SUCCESS!=RegCreateKey(HKEY\_CLASSES\_ROOT,L".seve",&hKey))

{

printf("创建子键失败!\n");

return 0;

}

else

{

printf("create .seve!\n");

}

RegCloseKey(hKey);

if(RegOpenKeyEx(HKEY\_CLASSES\_ROOT,L".arv",0,KEY\_ALL\_ACCESS,&hKey)!=ERROR\_SUCCESS)

{

printf("创建HKEY失败!\n");

return 0;

}

//set value

LPCWSTR szValueName1 = L"";

LPCWSTR szValueDate1=L"seven";

UINT cbLen=wcslen(szValueDate1);

if(RegSetValueExW(hKey,NULL,0,REG\_SZ,(const unsigned char \*)szValueDate1,cbLen\*2)==ERROR\_SUCCESS)

{

printf("set value!\n");

}

else

{

printf("创建REG\_SZ键值失败!\n");

return 0;

}

RegCloseKey(hKey);

2.创建处理键，并设置打开程序

//create opofile

if (ERROR\_SUCCESS!=RegCreateKey(HKEY\_CLASSES\_ROOT,L"opofile\\shell\\open\\command",&hKey))

{

printf("创建子键失败!\n");

return 0;

}

else

{

printf("create opofile!\n");

}

szValueDate1=L"\"F:\\SRC\\FILTER\\JUMP\\Debug\\JUMP.exe\" \"%1\"";

cbLen=wcslen(szValueDate1);

if(RegSetValueExW(hKey,NULL,0,REG\_SZ,(const unsigned char \*)szValueDate1,cbLen\*2)==ERROR\_SUCCESS)

{

printf("set opofile value!\n");

}

else

{

printf("创建REG\_SZ键值失败!\n");

return 0;

}

RegCloseKey(hKey);

getchar();