

送给初学者的礼物：C 游戏编程起源连载二 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/133/2021\\_2022\\_\\_E9\\_80\\_81\\_E7\\_BB\\_99\\_E5\\_88\\_9D\\_E5\\_c97\\_133522.htm](https://www.100test.com/kao_ti2020/133/2021_2022__E9_80_81_E7_BB_99_E5_88_9D_E5_c97_133522.htm) 第二章 使用Win32程序资源一、简介欢迎继续收看！通过本章题目可能你已经猜出了本章论题，我将教会你在Windows程序中使用资源。简单的讲，资源即数据，它们通常是和程序的EXE文件相关联的，但是它们又是独一无二的。首先，资源在运行过程中不能被修改。它们实际上都是只读文件，而且程序代码不能够直接访问它们。另外，资源并不在程序的数据区内。在装入时，程序资源通常在某个磁盘文件中，直到程序需要它们时才被装入。使用资源是一件很容易的事情，并且它的妙处无穷。Windows为我们提供了大量的资源类型，但我们这里只学一些最常用，最容易的：图标（icon）、光标（cursor）、位图（bitmap）、菜单（menu）和字符串（string）。此后，我还将教你建立自己风格类型的资源，使你为所欲为。重复一下，要想看懂本章，你得有点C语言的基础。C++有时用一用，但不影响你学习本章内容。并且我假定你已经读过了上一章内容“Windows编程基础”。还是用Microsoft Visual C++的编译器。资源脚本在进行细节之前，我们要先搞懂怎样要编译器知道它所要编译的资源类型。方法是使用称之为资源脚本的特殊文件，它是一个简单的文本文件，可以手工编辑，也可以让Visual C++自动编辑，或者你用其它的自动编辑器编辑。无论如何，资源脚本文件要有一个.rc的扩展名。大多数的脚本文件都从定义资源行开始，最简单的资源行通常要用到资源类型，就像这样：[identifier] [resource type]

[filename] 【标识符】 【资源类型】 【文件名称】 标识符可以用两种方式表示：一种是能表示资源意思的字符串，另一种是在资源相对应的头文件中用#define定义过的数字常量。如果你选择数字常量，这通常是一个好主意，别忘了把相应的头文件加入到你的资源脚本。资源脚本使用C语言风格的文件格式好像比较容易理解。以下是一个比较简单的资源脚本实例：

```
#include "resource.h"// iconsICON_MAIN ICON  
myicon.ico// bitmapsIMG_TILESET1 BITMAP  
tileset.bmpIMG_TILESET2 BITMAP tileset2.bmp
```

例子中的ICON\_MAIN和IMG\_TILESET是字符串呢，还是数字常量？这无伤大雅，编译器编译的时候会自己判断。如果发现在头文件中有#define的定义，那就认为是字符常量，否则，就是字符串。如果有些迷茫，不要紧。我将解释我们要用到的每一个资源类型。如果您觉得麻烦那让我们用全自动的资源插入系统吧！（在Visual C++中，在“插入”下拉菜单中，选择“资源”）现在你知道了建立资源脚本的基础知识，让我们开始进一步的行程吧！图标和光标 你每天在使用的大多数的Windows程序，都有自己的图标，简单的说，就是EXE文件同这个图标资源相关联了，独特风格的光标也是如此。你已经知道图标的脚本行样子了，光标的和它很相似，看看吧：

```
[identifier] CURSOR [filename][identifier] ICON [filename]
```

增加了一行脚本行后，也就是意味着你的EXE文件又多了一个关联。也就是说你的EXE文件要根据标识符去相应的位置寻找相应的文件[filename]。你可以使用任何你喜欢用的图标/光标编辑器去编辑相应的文件。我通常利用Visual C++中的编辑器。把资源脚本做出来后，并没有完事儿，因为你还不

道怎么调用相应的资源，要想知道图标和光标是怎样在你的程序中被调用的，让我们回过头来，看一看上一章中的窗口类（windows class）文件：

```
WNDCLASSEX sampleClass. //
declare structure variablesampleClass.cbSize =
sizeof(WNDCLASSEX).// always use this!sampleClass.style =
CS_DBLCLKS | CS_OWNDC | CS_HREDRAW |
CS_VREDRAW.// standard settingsampleClass.lpfWndProc =
MsgHandler. // message handler functionsampleClass.cbClsExtra =
0. // extra class info, not usedsampleClass.cbWndExtra = 0. // extra
window info, not usedsampleClass.hInstance = hInstance. //
parameter passed to WinMain()sampleClass.hIcon =
LoadIcon(NULL, IDI_WINLOGO).// Windows
logosampleClass.hCursor = LoadCursor(NULL, IDC_ARROW).//
standard cursorsampleClass.hbrBackground =
(HBRUSH)GetStockObject(BLACK_BRUSH). // a simple black
brushsampleClass.lpszMenuName = NULL. // no
menusampleClass.lpszClassName = "Sample Class" // class
namesampleClass.hIconSm = LoadIcon(NULL,
IDI_WINLOGO).// Windows logo again还记得它吧？这个hIcon
用来表示整个程序；hIconSm用来出现在开始菜单和窗口的标题
栏里；hCursor用来表示在你所创建的窗口中的光标的样子。
我向你保证，我们要实现自己的风格一点都不复杂。下面是
它们的原形：HICON LoadIcon( HINSTANCE hInstance, //
handle to application instance LPCTSTR lpIconName // icon-name
string or icon resource identifier).HCURSOR LoadCursor(
HINSTANCE hInstance, // handle to application instance
```

LPCWSTR lpCursorName // name string or cursor resource  
identifier). 100Test 下载频道开通，各类考试题目直接下载。详  
细请访问 [www.100test.com](http://www.100test.com)