1. INI文件

INI文件格式是某些平台或软件上的配置文件的非正式标准，以节(section)和键(key)构成，常用于微软Windows操作系统中。这种配置文件的文件扩展名多为INI，故名。

INI是英文“初始化”（initialization）的缩写。正如该术语所表示的，INI文件被用来对操作系统或特定程序初始化或进行参数设置。

文件格式编辑

节(section)

节用方括号括起来，单独占一行，例如：

[section]

键(key)

键(key)又名属性(property)，单独占一行用等号连接键名和键值，例如：

name=value

注释(comment)

注释使用英文分号（;）开头，单独占一行。在分号后面的文字，直到该行结尾都全部为注释，例如：

; comment text

1. 文件读写编辑
2. 文本编辑
3. 由于INI文件是纯文本格式，所以可以用任何纯文本编辑器来进行编辑其内容。
4. Windows API 函数读写
5. 读取
6. **DWORD** WINAPI GetPrivateProfileString(
7. \_In\_  **LPCTSTR** lpAppName,
8. \_In\_  **LPCTSTR** lpKeyName,
9. \_In\_  **LPCTSTR** lpDefault,
10. \_Out\_ **LPTSTR**  lpReturnedString,
11. \_In\_  **DWORD**   nSize,
12. \_In\_  **LPCTSTR** lpFileName
13. );
14. 参数说明：
15. lpAppName （输入参数）节名
16. lpKeyName （输入参数）键名
17. lpDefault （输入参数）默认值。INI读取失败时，得到这个值
18. lpReturnedString （输出参数）读取到的键值
19. nSize （输入参数）lpReturnedString的缓冲区大小
20. lpFileName （输入参数）INI文件的文件路径
21. 返回值：
22. 读取到的字符数，不包括末尾的'\0'字符或L'\0'字符。 [2]
24. 写入
25. **BOOL** WINAPI WritePrivateProfileString(
26. \_In\_ **LPCTSTR** lpAppName,
27. \_In\_ **LPCTSTR** lpKeyName,
28. \_In\_ **LPCTSTR** lpString,
29. \_In\_ **LPCTSTR** lpFileName
30. );
31. 参数说明：
32. lpAppName （输入参数）节名
33. lpKeyName （输入参数）键名
34. lpString （输入参数）需要写入的键值
35. lpFileName （输入参数）INI文件的文件路径
36. 返回值：
37. 写入成功返回非零值，写入失败返回零值，可以用GetLastError函数得到错误原因。
38. getcwd()
39. 函数原型：**char** \*getcwd( **char** \*buffer, **int** maxlen );
40. 功 能：获取当前工作目录
41. 参数说明：getcwd()会将当前工作目录的绝对路径复制到参数buffer所指的内存空间中,参数maxlen为buffer的空间大小。
42. 返 回 值：成功则返回当前工作目录，失败返回 FALSE。
43. 在某些 Unix 的变种下，如果任何父目录没有设定可读或搜索模式，即使当前目录设定了，getcwd()还是会返回 FALSE。有关模式与权限的更多信息见 chmod()。
44. 头文件：unistd.h（windows下为direct.h）
45. strcat()
46. 原型
47. **extern** **char** \*strcat(**char** \*dest, **const** **char** \*src);
48. 头文件
49. 在C中，函数原型存在 <string.h>头文件中。
50. 在C++中，则存在于<cstring>头文件中。
51. 功能
52. 把src所指字符串添加到dest结尾处(覆盖dest结尾处的'\0')。
53. 说明
54. src和dest所指内存区域不可以重叠且dest必须有足够的空间来容纳src的字符串。
55. 返回指向dest的指针。
56. GetPrivateProfileInt()
57. 为初始化文件中指定的条目获取它的整数值。
58. **UINT** WINAPI GetPrivateProfileInt
59. (
60. \_In\_LPCTSTR lpAppName, //The name of the section in the initialization file.
61. \_In\_LPCTSTR lpKeyName, //The name of the key whose value is to be retrieved.
62. \_In\_INT nDefault, //The default value to return if the key name cannot be found in the initialization file.
63. \_In\_LPCTSTR lpFileName //The name of the initialization file
64. );
65. 返回值编辑
66. Long，找到的条目的值；如指定的条目未找到，就返回默认值。如找到的数字不是一个合法的整数，函数会返回其中合法的一部分。如，对于“xyz=55zz”这个条目，函数返回55。这个函数也能理解采用标准C语言格式的十六进制数字：用0x作为一个十六进制数字的前缀——所以0x55ab等价于vb的&H55AB
67. 参数表编辑
68. lpApplicationName String，指定在其中查找条目的小节。注意这个字串是不区分大小写的
69. lpKeyName String，欲获取的设置项或条目。这个支持不区分大小写
70. nDefault Long，指定条目未找到时返回的默认值
71. lpFileName String，初始化文件的名字。如果没有指定完整的路径名，windows就会在Windows目录中搜索文件
72. strstr(str1,str2)

strstr(str1,str2) 函数用于判断字符串str2是否是str1的子串。如果是，则该函数返回str2在str1中首次出现的地址；否则，返回NULL。

1. WSAStartup()
2. **int** WSAStartup( **WORD** wVersionRequested, LPWSADATA lpWSAData);
3. 为了在应用程序当中调用任何一个Winsock API函数，首先第一件事情就是必须通过WSAStartup函数完成对Winsock服务的初始化，因此需要调用WSAStartup函数。使用Socket的程序在使用Socket之前必须调用WSAStartup函数。该函数的第一个参数指明程序请求使用的Socket版本，其中高位字节指明副版本、低位字节指明主版本；操作系统利用第二个参数返回请求的Socket的版本信息。当一个应用程序调用WSAStartup函数时，操作系统根据请求的Socket版本来搜索相应的Socket库，然后绑定找到的Socket库到该应用程序中。以后应用程序就可以调用所请求的Socket库中的其它Socket函数了。
4. Error lnk1112:模块计算机类型“X86”与目标计算机“X64”冲突