文章
姚 鑫 · 七月 13, 2021 阅读大约需 6 分钟

第二章 使用文件和目录属性和属性

第二章 使用文件和目录属性和属性

%Library.File类还提供了许多类方法，可以使用这些方法来获取有关文件和目录的信息，或者查看或设置它们的属性和属性。

检查文件和目录是否存在

要确定给定文件是否存在，请使用Existes()方法并指定文件名作为参数。例如：

```
DHC-APP>write ##class(%File).Exists("e:\temp\")
1
```

同样，要找出给定目录是否存在，请使用DirectoryExists()方法，并将该目录指定为参数。例如：

```
DHC-APP>write ##class(%File).DirectoryExists("e:\temp")
1
```

如前所述，这些方法在Unix上将文件或目录名视为区分大小写，但在Windows上不区分大小写。此外，如果指定部分文件名或目录名，则该方法引用的文件或目录相对于包含正在使用的命名空间的默认全局数据库的目录。例如：

```
DHC-APP>write ##class(%File).Exists("cache.dat")
1
```

查看和设置文件和目录权限

%Library.File类提供了许多类方法，可以使用这些方法查看或设置文件或目录的权限。

查看文件或目录是只读的还是可写的

给定文件或目录名，如果文件或目录是只读的，ReadOnly()方法返回1，否则返回0:

```
DHC-APP>write ##class(%File).ReadOnly("export.xml")
1
DHC-APP>write ##class(%File).ReadOnly("E:\temp")
0
```

同样，给定一个文件或目录名，如果该文件或目录可写，则Writeable()方法返回1，否则返回0:

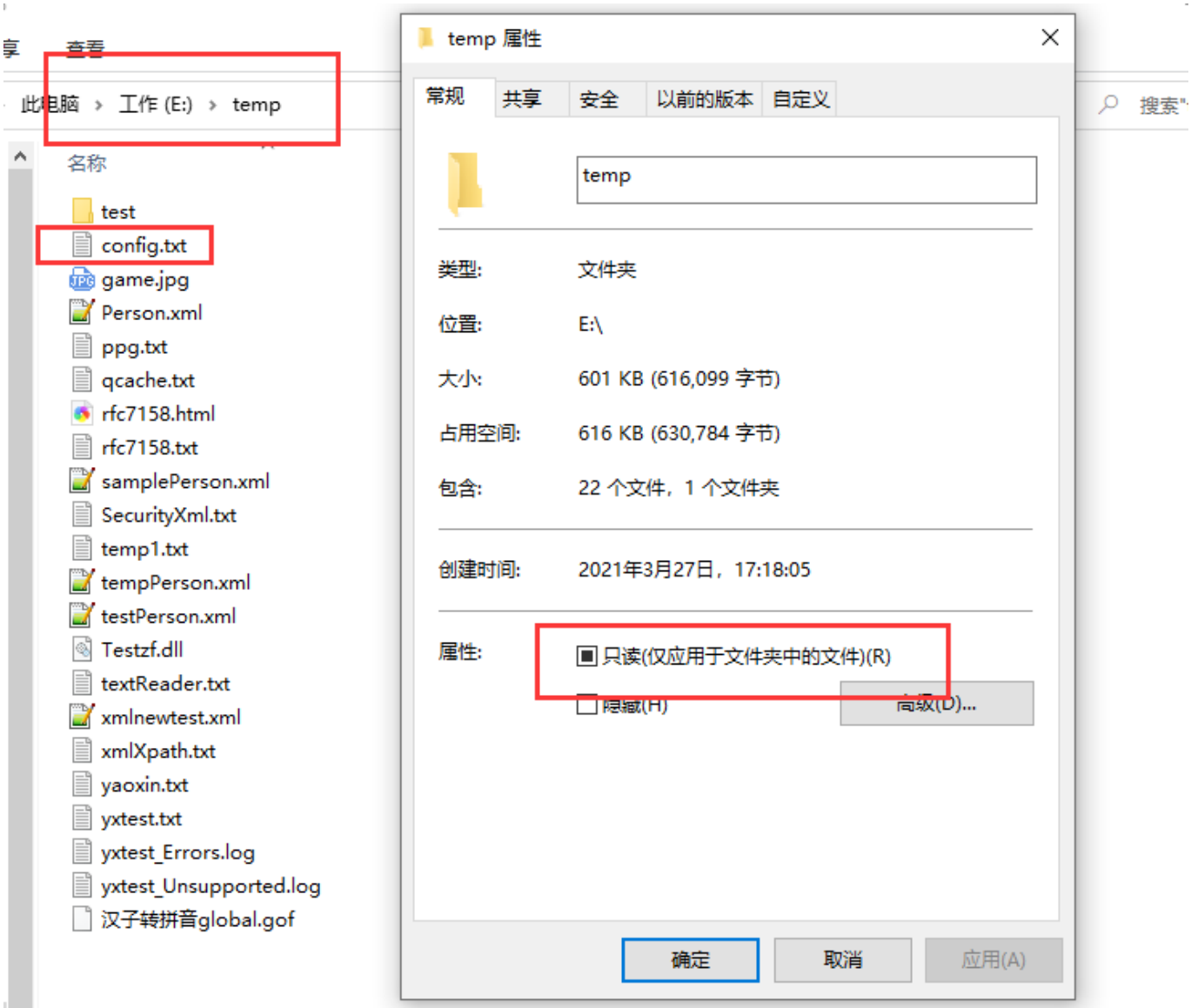
```
DHC-APP>write ##class(%File).Writeable("export.xml")
0
DHC-APP>write ##class(%File).Writeable("e:\temp")
1
```

将文件或目录设为只读或可写(窗口)

要使Windows上的文件或目录成为只读的，请使用SetReadOnly()方法，该方法返回一个布尔值来指示成功或失败。此方法采用三个参数，第二个参数在Windows中被省略。第一个参数是文件或目录的名称。第三个参数是输出参数。如果为负，它包含操作系统返回的错误代码，以防方法失败。

在下面的示例中，对SetReadOnly()的调用成功地将文件"e:\temp\config.txt更改为只读。

```
DHC-APP>write ##class(%File).ReadOnly("e:\temp\config.txt")
0
DHC-APP>write ##class(%File).SetReadOnly("e:\temp\config.txt",,.return)
1
DHC-APP>write ##class(%File).ReadOnly("e:\temp\config.txt")
```



在下面的示例中，对SetReadOnly()的调用失败，Windows系统错误代码为5，这意味着“ 访问被拒绝。 ”

```
DHC-APP>write ##class(%File).SetReadOnly("C:\",,.return)
0
DHC-APP>w return
-5
```

要使Windows上的文件或目录可写，请使用SetWriteable()方法。此方法采用相同的三个参数，第二个参数在Windows中再次被省略。

```
DHC-APP>write ##class(%File).Writeable("e:\temp\config.txt")
0
DHC-APP>write ##class(%File).SetWriteable("e:\temp\config.txt",,.return)
1
DHC-APP>write ##class(%File).Writeable("e:\temp\config.txt")
1
```

将文件或目录设为只读或可写(Unix)

在Unix上，也可以使用SetReadOnly()和SetWriteable()方法，但是由于第二个参数的存在，它们的行为有些不同。

但是，在Unix中，为所有者、组和用户指定不同的权限。要更好地控制文件和目录权限，请参阅查看或设置文件和目录属性一节。

查看和设置文件和目录属性

要在更详细的级别查看或设置文件或目录的属性，请使用%Library.File的Attributes()和SetAttributes()方法。文件属性由集中表示为整数的位序列表示。各个位的含义取决于底层操作系统。

查看文件和目录属性

%Library.File的Attributes()方法需要文件名或目录名作为参数，并返回以整数表示的属性位序列。

以下示例在Windows系统上运行：

```
DHC-APP>write ##class(%File).Attributes("cache.dat")
32
DHC-APP>write ##class(%File).Attributes("e:\temp")
16
DHC-APP>write ##class(%File).Attributes("secret.zip")
35
DHC-APP>write ##class(%File).Attributes("e:\temp\config.txt")
32
```

在第一个例子中，32表示cache.dat是一个存档文件。在第二个例子中，16表示C:\temp是一个目录。在第三个示例中，设置了不止一个位，35表示secret.zip是隐藏的(2)和只读的(1)存档(32)。加32 + 2 + 1 = 35。

以下示例在Unix系统上运行:

```
write ##class(%File).Attributes("/home")
16877
```

在本例中，16877表示/home是一个目录(16384)，拥有者拥有读取(256)、写入(128)和执行(64)权限；读取(32)和执行(8)组权限；并为他人读取(4)和执行(1)权限。加上16384+256+128+64+32+8+4+1 = 16877。

设置文件和目录属性

相反，SetAttributes()方法设置文件或目录的属性，并返回一个布尔值来指示成功或失败。这个方法需要三个参数。第一个参数是文件或目录的名称。第二个参数是一个整数，表示文件或目录具有的所需属性。第三个参数是输出参数。如果为负，它包含操作系统返回的错误代码，以防方法失败。

以下示例在Windows上，通过设置1位，使文件e:\temp\config.txt成为只读文件:

```
DHC-APP>write ##class(%File).Attributes("e:\temp\config.txt")
32
DHC-APP>write ##class(%File).SetAttributes("e:\temp\config.txt",33,.return)
1
DHC-APP>write ##class(%File).Attributes("e:\temp\config.txt")
33
```

以下示例在Unix上，将默认目录中文件myfile的权限从644更改为完全权限(777):

```
USER>write ##class(%File).Attributes("myfile")
33188
USER>write ##class(%File).SetAttributes("myfile",33279,.return)
1
USER>write ##class(%File).Attributes("myfile")
33279
```

通过将常规文件(32768)的值与所有者(448)、组(56)和其他人(7)的掩码相加来计算期望的属性值。

查看其他文件和目录属性

%Library.File的其他类方法允许检查文件和目录的各种其他属性。

GetFileDateCreated()方法返回以$H格式创建文件或目录的日期:

```
DHC-APP>write $zdate(##class(%File).GetFileDateCreated("stream"))
04/11/2020
DHC-APP>write $zdate(##class(%File).GetFileDateCreated("e:\temp"))
03/27/2021
```

注意:Windows是目前唯一跟踪实际创建日期的平台。其他平台存储最后一次文件状态更改的日期。

GetFileDateModified()方法以$H格式返回文件或目录的修改日期:

```
DHC-APP>write $zdate(##class(%File).GetFileDateModified("cache.dat"))
07/11/2021
```

方法返回文件的大小，以字节为单位:

```
DHC-APP>write ##class(%File).GetFileSize("e:\temp\config.txt")
220
```

GetDirectorySpace()方法返回驱动器或目录中的可用空间量和总空间。根据第四个参数(可以是0、1或2)的值，空间可以以字节、MB(默认值)或GB为单位返回。在本例中，2表示空间以GB为单位返回:

```
DHC-APP>set status = ##class(%File).GetDirectorySpace("C:", .FreeSpace, .TotalSpace,
2)
DHC-APP>write FreeSpace
128.83
DHC-APP>write TotalSpace
247.97
```

在Windows上，如果将目录名传递给此方法，返回的空间量就是整个驱动器的空间量。

对于GetDirectorySpace()方法，返回的任何错误状态都是操作系统级错误。在下面的示例中，Windows系统错误代码3表示 “ 系统找不到指定的路径。 ”

```
DHC-APP>set status = ##class(%File).GetDirectorySpace("Q:", .FreeSpace, .TotalSpace,
2)
DHC-APP>do $system.Status.DisplayError(status)

?? #83: ????=3
```

#Caché

源 URL: https://cn.community.intersystems.com/post/%E7%AC%AC%E4%BA%8C%E7%AB%A0-%E4%BD%BF%E7%94%A8%E6%96%87%E4%BB%B6%E5%92%8C%E7%9B%AE%E5%BD%95%E5%B1%9E%E6%80%A7%E5%92%8C%E5%B1%9E%E6%80%A7