文章
姚 鑫 · 七月 15, 2021 阅读大约需 4 分钟

第四章 使用目录

第四章 使用目录

使用目录

%Library。File类提供了几个类方法，允许对目录执行各种操作。

创建目录

若要创建目录，请使用CreateDirectory()方法，该方法返回一个布尔值来指示成功或失败。这个方法需要两个参数。第一个参数是要创建的目录的名称。第二个参数是输出参数。如果为负，它包含操作系统返回的错误代码，以防方法失败。

如果e:\temp已经存在，以下命令将失败，并显示Windows系统错误代码183，这意味着 “ 当文件已经存在时，无法创建该文件。 ”

```
DHC-APP>write ##class(%File).CreateDirectory("e:\temp", .return)
0
DHC-APP>w return
-183
```

如果e:\temp已经存在，但e:\temp\test不存在，则以下命令会失败，因为CreateDirectory()最多创建给定目录路径中的最后一个目录。所以返回的Windows系统错误代码是3，或者 “ 系统找不到指定的路径。 ”

```
DHC-APP>write ##class(%File).CreateDirectory("e:\temp\test\this", .return)
0
DHC-APP>w return
-183
```

以下示例使用Windows系统代码0或 “ 操作成功完成 ” 成功完成

```
DHC-APP>write ##class(%File).CreateDirectory("e:\temp\test\this", .return)
1
DHC-APP>w return
0
```

类似的CreateNewDir()方法在指定的父目录中创建新目录。这个方法需要三个参数。第一个参数是父目录的名称。第二个参数是要创建的目录的名称。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。

下面的第一个示例在父目录e:\temp中创建了一个名为newdir的目录。第二个示例在默认目录中创建了一个名为newdir的新目录。

```
DHC-APP>write ##class(%File).CreateNewDir("", "newdir", .return)
1
DHC-APP>write ##class(%File).CreateNewDir("e:\temp", "newdir", .return)
1
```

另一个相关的方法，CreateDirectoryChain()，创建给定目录路径上的所有目录(如果可能)。

下面的第一个示例在父目录C:\temp中创建了三个嵌套目录。第二个示例在默认目录中创建三个嵌套目录。

```
DHC-APP>write ##class(%File).CreateDirectoryChain("e:\temp\one\two\three", .return)
1
DHC-APP>write ##class(%File).CreateDirectoryChain("one\two\three", .return)
1
```

复制目录

要复制目录，请使用CopyDir()方法，该方法返回一个布尔值来指示成功或失败。

此方法采用五个参数:

1. pSource源—指定源目录的名称。
2. pTarget 目标—指定目标目录的名称。
3. pOverlay —指定是否覆盖目标目录(如果存在)。默认值为0。
4. pCreated —输出参数，包含复制过程中创建的文件或目录的数量。
5. pDeleteBeforeCopy —指定在执行复制之前是否删除目标目录中存在的任何文件。默认值为0。

pSource或pTarget的部分目录名是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计算的。

与目录创建方法不同，CopyDir()没有返回系统错误代码的输出参数。

在下面的第一个示例中，复制操作成功，46个文件和目录从e:\temp复制到e:\temp2。在第二个示例中，复制操作成功，46个文件和目录从e:\temp复制到默认目录中的目录temp2。

```
DHC-APP>write ##class(%File).CopyDir("e:\temp", "e:\temp2", 0, .pCreated, 0)
1
DHC-APP>write pCreated
29
```

在下面的最后一个示例中，pOverlay被设置为0，因此复制失败，因为目标目录已经存在。

```
DHC-APP>write ##class(%File).CopyDir("e:\temp", "e:\temp2", 0, .pCreated, 0)
0
DHC-APP>write pCreated
0
```

删除目录

若要删除非空目录，请使用RemoveDirectory()方法，该方法成功时返回1，失败时返回0。这个方法需要两个参数。

第一个参数是要删除的目录的名称。第二个参数是输出参数，包含操作系统返回的错误代码，以防方法失败。

在下面的第一个示例中，方法成功了。第二个示例失败，出现Windows错误代码145，或“ 目录不为空 ”.

```
DHC-APP>write ##class(%File).RemoveDirectory("e:\temp2\newdir", .return)
1
DHC-APP>write ##class(%File).RemoveDirectory("e:\temp2", .return)
0
DHC-APP>write return
-145
```

若要删除目录，包括任何子目录，请使用RemoveDirectoryTree()方法，该方法成功时返回1，失败时返回0。与RemoveDirectory()方法不同，RemoveDirectoryTree()没有输出参数来返回系统错误代码。

即使目录和任何子目录不为空，RemoveDirectoryTree()也会成功。

```
DHC-APP>write ##class(%File).RemoveDirectoryTree("e:\temp2")
1
```

重命名目录

若要重命名目录，请使用rename()方法，该方法成功时返回1，失败时返回0。这个方法需要三个参数。第一个参数是要重命名的目录的名称，第二个参数是新名称。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。

仅当目录与正在处理的文件系统相同时，使用Rename()重命名目录才有效。

在下面的第一个示例中，方法成功了。在第二个示例中，"e:\temp\one"不存在，因此该方法失败，出现Windows错误代码3，或者“ 系统找不到指定的路径。 ”

```
DHC-APP>write ##class(%File).Rename("e:\temp\one", "e:\temp\newone", .return)   1
DHC-APP>write ##class(%File).Rename("e:\temp\one", "e:\temp\newone", .return)
0
DHC-APP>write return
-2
```

使用此方法时，请小心指定路径，因为下面的示例会将e:\temp\one移动到默认目录，然后将其重命名为newname。

```
DHC-APP>write ##class(%File).Rename("e:\temp\newone", "newname", .return)
1
```

#Caché

源URL:
https://cn.community.intersystems.com/post/%E7%AC%AC%E5%9B%9B%E7%AB%A0-%E4%BD%BF%E7%94%A8%E7%9B%AE%E5%BD%95