
文章

[姚鑫](#) · 七月 17, 2021 阅读大约需 4 分钟

第六章 使用%File对象

第六章 使用%File对象

如果想要操作文件本身，需要使用%Library.File的%New()方法实例化%File对象。该类还提供了允许使用该文件的实例方法。

注意:本节提供了几个使用%File对象的示例，以供说明。

对于简单的文件读写，使用%Stream.FileCharacter和%Stream.FileBinary。因为它们提供了额外的功能，例如，以正确的模式自动打开文件。

创建%File对象的实例

要使用文件，需要使用%New()方法实例化表示该文件的%File对象。该文件可能已经存在，也可能不存在于磁盘上。

以下示例在默认目录中为文件export.xml实例化一个%File对象。

```
set fileObj = ##class(%File).%New("export.xml")
```

打开和关闭文件

实例化%File对象后，需要使用open()方法打开文件，以读取或写入该文件:

```
USER>set status = fileObj.Open()
```

```
USER>write status  
1
```

使用Close()方法关闭文件:

```
USER>do fileObj.Close()
```

检查%File对象的属性

一旦实例化了文件，就可以直接检查文件的属性。

```
USER>write fileObj.Name  
export.xml  
USER>write fileObj.Size  
2512
```

```
USER>write $zdate(fileObj.DateCreated)
11/18/2020
USER>write $zdate(fileObj.DateModified)
11/18/2020
USER>write fileObj.LastModified
2020-11-18 14:24:38
USER>write fileObj.IsOpen
0
```

请注意，LastModified是人类可读的时间戳，而不是\$H格式的日期。

属性“大小Size”、“创建日期DateCreated”、“修改日期DateModified”和“最后修改日期LastModified”是在访问时计算的。为不存在的文件访问这些属性会返回-2，表示找不到该文件。

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。

```
USER>write ##class(%File).Exists("foo.xml")
0
USER>set fooObj = ##class(%File).%New("foo.xml")

USER>write fooObj.Size
-2
```

如果文件已打开，可以通过访问CanonicalName属性来查看其规范名称，这是根目录的完整路径。

```
USER>write fileObj.CanonicalName

USER>set status = fileObj.Open()

USER>write fileObj.IsOpen
1
USER>write fileObj.CanonicalName
c:\intersystems\IRIS\mgr\user\export.xml
```

从文件中读取

要读取文件，可以打开文件，然后使用Read()方法。

以下示例读取messages.log的前200个字符。

```
USER>set messages = ##class(%File).%New(##class(%File).ManagerDirectory() _ "messages.log")

USER>set status = messages.Open("RU")

USER>write status
1
USER>set text = messages.Read(200, .sc)

USER>write text
```

```
*** Recovery started at Mon Dec 09 16:42:01 2019
    Current default directory: c:\intersystems\IRIS\mgr
    Log file directory: .\
    WIJ file spec: c:\intersystems\IRIS\mgr\IR
USER>write sc
1
USER>do messages.Close()
```

要从文件中读取整行，请使用ReadLine()方法，该方法继承自%Library.File的父类%Library.AbstractStream。

下面的示例读取E:\temp\new.txt的第一行。

```
/// desc: ????
/// w ##class(Demo.FileDemo).ReadFileData("E:\temp\new.txt")
ClassMethod ReadFileData(str)
{
    s fileObj = ##class(%File).%New(str)
    s status = fileObj.Open("RU")
    w status,!
    s text = fileObj.ReadLine(,.sc)
    w text,!
    w sc,!
    d fileObj.Close()

    q ""
}
```

写入文件

要写入文件，可以打开文件，然后使用Write()或WriteLine()方法。

以下示例将一行文本写入新文件。

```
/// desc: ????
/// w ##class(Demo.FileDemo).WriteFileData("E:\temp\new.txt")
ClassMethod WriteFileData(str)
{
    s fileObj = ##class(%File).%New(str)
    s status = fileObj.Open("RUWSN")
    w status,!
    s status = fileObj.WriteLine("Writing to a new file.")
    w status,!
    w fileObj.Size,!
    d fileObj.Rewind()
    s text = fileObj.ReadLine(,.sc)
    w text,!
    q ""
}
```

倒回文件

从文件读取或写入文件后，希望使用Rewind()方法倒回文件，以便可以从文件开头执行操作。

从上一个示例停止的地方开始，fileObj现在位于其末尾。倒回文件并再次使用WriteLine()会覆盖该文件。

```
USER>set status = fileObj.Rewind()

USER>write status
1
USER>set status = fileObj.WriteLine("Rewriting the file from the beginning.")

USER>write status
1
USER>write fileObj.Size
40
```

关闭文件并重新打开它也会倒回文件。

```
USER>do fileObj.Close()

USER>set status = fileObj.Open("RU")

USER>write status
1
USER>set text = fileObj.ReadLine(, .sc)

USER>write sc
1
USER>write text
Rewriting the file from the beginning.
```

清除文件

要清除文件，可以打开文件，然后使用Clear()方法。这将从文件系统中删除该文件。

以下示例清除默认目录中的junk.xml。

```
USER>write ##class(%File).Exists("junk.xml")
1
USER>set fileObj = ##class(%File).%New("junk.xml")

USER>set status = fileObj.Open()

USER>write status
1
USER>set status = fileObj.Clear()

USER>write status
1
USER>write ##class(%File).Exists("junk.xml")
0
```

[#Caché](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%AC%AC%E5%85%AD%E7%AB%A0-%E4%BD%BF%E7%94%A8file%E5%AF%B9%E8%B1%A1>