

記事

[Megumi Kakechi](#) · 2023年2月6日 4m read

ファイル操作関連のTips

ファイルとディレクトリの操作に使用できる API、[%Library.File](#) クラス (%File が省略形) の便利なサンプルコードをご紹介します。

こちらのAPIを使用して、ディレクトリやファイルの中身をのぞいたり、作成・書込み・読み込みを行ってみます。

[ディレクトリ内のファイル一覧をリストする](#)

以下のサンプルは、C:\temp フォルダ内の *.txt ファイルをリストします(ファイルパスサイズ作成日)。

```
set stmt = ##class(%SQL.Statement).%New()  
set status = stmt.%PrepareClassQuery("%File", "FileSet")  
set rs = stmt.%Execute("c:\temp", "*.txt")  
while rs.%Next(.rStatus) { write !, rs.%Get("Name")_"_"_rs.%Get("Size")_"byte_"  
"_"_rs.%Get("DateCreated") }
```

サブディレクトリ内のファイルを含む場合はこちら (filetest.mac で保存)

```
test(dir)  
do disp(dir)  
quit  
disp(dir) {  
set rs=##class(%ResultSet).%New("%File:FileSet")  
do rs.Execute(dir)  
while(rs.Next()) {  
set file=rs.Get("Name")  
if rs.Get("Type")="D" { // ???????? (Type="D":?????)???  
do disp(file)  
}  
else {  
write file,!  
}  
}  
quit  
}
```

実行結果：

```
USER>do ^filetest("c:\temp\java")  
C:\temp\java\IRISPersentG.class  
C:\temp\java\IRISPersentG.java  
C:\temp\java\Sample\Address.java  
C:\temp\java\Sample\Person.java  
C:\temp\java\SLISTTestIRIS.class  
C:\temp\java\SLISTTestIRIS.java
```

シンプルなファイル作成 & 書き込み & 読み込み

作成 & 書き込み

```
set file=##class(%File).%New("c:\temp\file.txt")
do file.Open("WSN") // ??????????????OPEN????????(?)? "WSNK\UTF8" ??????
do file.WriteLine("This is a test.")
do file.WriteLine("????????")
do file.Close()
kill file
```

OPEN モード・パラメータ

または (Streamの場合)、

```
set stream=##class(%Stream.FileCharacter).%New()
set stream.TranslateTable="SJIS"
set stream.Filename="C:\temp\file2.txt"
do stream.WriteLine("This is a test2.")
do stream.WriteLine("????????")
do stream.%Save()
kill stream
```

読み込み

```
set file=##class(%File).%New("C:\temp\file.txt")
do file.Open("R")
while 'file.AtEnd { write !,file.ReadLine()}
// file.Read() ????????32000????????.Read(len) ??????????
do file.Close()
kill file
```

または (Streamの場合)、

```
Set stream=##class(%Stream.FileCharacter).%New()
set stream.TranslateTable="SJIS"
Set sc=stream.LinkToFile("c:\temp\file2.txt")
while 'stream.AtEnd { write !,stream.ReadLine() }
kill stream
```

ファイルをバイト単位で読み込む

%Stream.FileCharacter の場合、Readした時点でUnicodeに変換されているため、バイト数で切ことは難しくなります（文字数単位となる）。

バイト単位で読み込みたい場合は、FileBinaryを使い「無変換」で取り込み、それを Unicode に手動で変換する方法を使用します。

```
/// 4byte?????
set stream=##class(%Stream.FileBinary).%New()
do stream.LinkToFile("C:\temp\file.txt")
while 'stream.AtEnd {
```

```
set line=stream.ReadLine(4)
set x=$ZCVT(line,"I","SJIS")
write x,!
}
```

注意

バイト単位で読む場合、[\\$ZCVT](#)

で変換できない(日本語文字区切りで2バイト目が読まれていない)パターンがでてくるので注意が必要です。
その場合は、\$ZCVT の第4引数 handle オプションを使います。

```
/// handle ??????????????????
set handle=""
set stream=##class(%Stream.FileBinary).%New()
do stream.LinkToFile("C:\temp\file.txt")
while 'stream.AtEnd {
  set line=stream.ReadLine(3)
  set x=$ZCVT(line,"I","SJIS",handle)
  write x,!
}
```

ファイルの存在有無をチェック

```
write ##class(%File).Exists("C:\temp\file.txt")    // 1:???? , 0:?????
```

ディレクトリの有無をチェック

```
write ##class(%File).DirectoryExists("C:\temp")    // 1:???? , 0:?????
```

ファイルサイズを取得(ファイル単位)

```
set file="C:\temp\file.txt"
write ##class(%File).GetFileSize(file)             // bytes
```

ファイルパスからファイル名部分だけ抜き出す

```
Set filename=##class(%File).GetFilename("c:\work\isj\test.xml") // Windows
Write filename    // test.xml
Set filename=##class(%File).GetFilename("/root/work/test.xml")  // Unix
Write filename    // test.xml
```

他にも便利な使用方法がありますので、詳細は以下のドキュメントをご覧ください。

[%Library.File の使用方法](#)



ご参考

[ObjectScript クックブック](#)

[ファイル入出力処理をスクリプトで記述する方法](#)

[#ObjectScript](#) [#ヒントとコツ](#) [#Caché](#) [#Ensemble](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

ソースURL:

<https://jp.community.intersystems.com/post/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB%E6%93%8D%E4%BD%9C%E9%96%A2%E9%80%A3%E3%81%AEtips>