# FileMode

指定操作系统打开文件的方式

### 一．FileMode.Append

**Append**追加：

如果文件存在，则打开文件，把指针指到文件的末尾；如果不存在，则新建文件

注意：

1. 这需要 [Append](https://docs.microsoft.com/zh-cn/dotnet/api/system.security.permissions.fileiopermissionaccess?view=netframework-4.7.2#System_Security_Permissions_FileIOPermissionAccess_Append) 权限。
2. FileMode.Append 只能与 FileAccess.Write 一起使用。
3. 试图查找文件尾之前的位置时会引发 [IOException](https://docs.microsoft.com/zh-cn/dotnet/api/system.io.ioexception?view=netframework-4.7.2) 异常，并且任何试图读取的操作都会失败并引发 [NotSupportedException](https://docs.microsoft.com/zh-cn/dotnet/api/system.notsupportedexception?view=netframework-4.7.2) 异常。

### 二．FileMode.Create

**Create**：新建

1. 如果文件存在，则覆盖原有文件，把指针指到文件的开始，文件的创建日期会更新，这需要 [Write](https://docs.microsoft.com/zh-cn/dotnet/api/system.security.permissions.fileiopermissionaccess?view=netframework-4.7.2#System_Security_Permissions_FileIOPermissionAccess_Write) 权限。
2. 如果文件不存在，则使用 [CreateNew](https://docs.microsoft.com/zh-cn/dotnet/api/system.io.filemode?view=netframework-4.7.2#System_IO_FileMode_CreateNew)；否则使用 [Truncate](https://docs.microsoft.com/zh-cn/dotnet/api/system.io.filemode?view=netframework-4.7.2#System_IO_FileMode_Truncate)。
3. 如果该文件已存在但为隐藏文件，将引发 [UnauthorizedAccessException](https://docs.microsoft.com/zh-cn/dotnet/api/system.unauthorizedaccessexception?view=netframework-4.7.2)异常。

### 三．FileMode.CreateNew

**CreateNew**：新建新的文件

1. 指定操作系统应创建新文件。 这需要 [Write](https://docs.microsoft.com/zh-cn/dotnet/api/system.security.permissions.fileiopermissionaccess?view=netframework-4.7.2#System_Security_Permissions_FileIOPermissionAccess_Write) 权限。
2. 如果文件已存在，则将引发 [IOException](https://docs.microsoft.com/zh-cn/dotnet/api/system.io.ioexception?view=netframework-4.7.2)异常。

### 四．FileMode.OpenOrCreate

**OpenOrCreate**：打开或是新建

如果文件存在，则打开文件，把指针指到文件的开始；如果文件不存在，则新建文件

注意：

1. 如果用 FileAccess.Read 打开文件，则需要 [Read](https://docs.microsoft.com/zh-cn/dotnet/api/system.security.permissions.fileiopermissionaccess?view=netframework-4.7.2#System_Security_Permissions_FileIOPermissionAccess_Read)权限。
2. 如果文件访问为 FileAccess.Write，则需要 [Write](https://docs.microsoft.com/zh-cn/dotnet/api/system.security.permissions.fileiopermissionaccess?view=netframework-4.7.2#System_Security_Permissions_FileIOPermissionAccess_Write)权限。
3. 如果用 FileAccess.ReadWrite 打开文件，则同时需要 [Read](https://docs.microsoft.com/zh-cn/dotnet/api/system.security.permissions.fileiopermissionaccess?view=netframework-4.7.2#System_Security_Permissions_FileIOPermissionAccess_Read) 和 [Write](https://docs.microsoft.com/zh-cn/dotnet/api/system.security.permissions.fileiopermissionaccess?view=netframework-4.7.2#System_Security_Permissions_FileIOPermissionAccess_Write)权限。

### 五. FileMode.Truncate

1. 指定操作系统应打开现有文件。 该文件被打开时，内容将被清除，把指针指到文件开头。 这需要 [Write](https://docs.microsoft.com/zh-cn/dotnet/api/system.security.permissions.fileiopermissionaccess?view=netframework-4.7.2#System_Security_Permissions_FileIOPermissionAccess_Write) 权限。

2. 尝试从使用 FileMode.Truncate 打开的文件中进行读取将导致 [ArgumentException](https://docs.microsoft.com/zh-cn/dotnet/api/system.argumentexception?view=netframework-4.7.2) 异常。

### 六. FileMode.Open

1. Open：打开文件

2. 如果文件不存在，引发一个 [FileNotFoundException](https://docs.microsoft.com/zh-cn/dotnet/api/system.io.filenotfoundexception?view=netframework-4.7.2) 异常。

### 七.示例

以下FileStream构造函数将打开一个现有文件 (FileMode.Open)。

FileStream s2 = new FileStream(name, FileMode.Open, FileAccess.Read, FileShare.Read);