FileOpen

Opens an existing file for reading and writing.

> **Note:** To prevent memory leaks, ensure each successful call to fileOpen has a matching call to fileClose.

> **Tip:** The file functions should not be used to implement configuration files. It is encouraged to use the XML functions for this instead.

> ⚠️ **Warning:** As of 1.5.4 r10413, this function will fail when trying to access a script file of another resource, even with *general.ModifyOtherObjects* rights granted, which uses a *mysql* connection through dbConnect when *database_credentials_protection* is enabled in the server configuration. Additionally, meta.xml will be un-writable and will always open in read-only mode.

Syntax

```
file fileOpen ( string filePath [, bool readOnly = false ] )
```

OOP Syntax Help! I don't understand this!

> **Note**: *The function will only attempt to open the file, it won't create it.*
> **Method**: *File(...)*

ADDED/UPDATED IN VERSION 1.5.6 r11865:

OOP Syntax Help! I don't understand this!

> **Note**: *This is a static function underneath the File class. Using **File(...)** to open a file will attempt to create the file, if it doesn't exist*
> **Method**: *File.open(...)*

Required Arguments

- **filePath:** The filepath of the file in the following format: **":resourceName/path"**. 'resourceName' is the name of the resource the file is in, and 'path' is the path from the root directory of the resource to the file.

  For example, if there is a file named 'coolObjects.txt' in the resource 'objectSearch', it can be opened from another resource this way: *fileOpen(":objectSearch/coolObjects.txt")*.
  If the file is in the current resource, only the file path is necessary, e.g. *fileOpen("coolObjects.txt")*.

Optional Arguments

- **readOnly:** By default, the file is opened with reading and writing access. You can specify *true* for this parameter if you only need reading access.

Returns

If successful, returns a file handle for the file. Otherwise returns *false* (f.e. if the file doesn't exist).