

## FolkerKinzel.Helpers Namespace

Helper classes

### Classes

<a href="#"><b>BinaryFile</b></a>	<b>Helper class for operations with binary files.</b>
<a href="#">TextFile</a>	Helper class for operations with text files.

## BinaryFile Class

Helper class for operations with binary files.

### Definition

**Namespace:** [FolkerKinzel.Helpers](#)

**Assembly:** FolkerKinzel.Helpers (in FolkerKinzel.Helpers.dll) Version: 1.0.0+58110188496bad6fb659c326e00a01796f03a8ac

**C#**

```
public static class BinaryFile
```

**Inheritance**     [Object](#) → BinaryFile

### Methods

<a href="#">Load</a>	Loads a binary file.
<a href="#">Save</a>	Saves a byte array to a file.

### See Also

#### Reference

[FolkerKinzel.Helpers Namespace](#)

## BinaryFile.Load Method

Loads a binary file.

### Definition

**Namespace:** [FolkerKinzel.Helpers](#)

**Assembly:** FolkerKinzel.Helpers (in FolkerKinzel.Helpers.dll) Version: 1.0.0+58110188496bad6fb659c326e00a01796f03a8ac

#### C#

```
public static byte[] Load(  
    string filePath  
)
```

### Parameters

*filePath* [String](#)

The file path.

### Return Value

[Byte\[\]](#)

The content of the binary file as byte array.

### Exceptions

<a href="#">ArgumentNullException</a>	<i>filePath</i> is null.
<a href="#">ArgumentException</a>	<i>filePath</i> is not a valid file path.
<a href="#">IOException</a>	I/O error.

### See Also

#### Reference

[BinaryFile Class](#)

[FolkerKinzel.Helpers Namespace](#)

## BinaryFile.Save Method

Saves a byte array to a file.

### Definition

**Namespace:** [FolkerKinzel.Helpers](#)

**Assembly:** FolkerKinzel.Helpers (in FolkerKinzel.Helpers.dll) Version: 1.0.0+58110188496bad6fb659c326e00a01796f03a8ac

```
C#  
  
public static void Save(  
    string filePath,  
    byte[] bytes  
)
```

### Parameters

*filePath* [String](#)

The file path.

*bytes* [Byte\[\]](#)

The byte array to save.

### Exceptions

<a href="#">ArgumentNullException</a>	<i>filePath</i> or <i>bytes</i> is null.
<a href="#">ArgumentException</a>	<i>filePath</i> is not a valid file path.
<a href="#">IOException</a>	I/O error.

### See Also

#### Reference

[BinaryFile Class](#)

[FolkerKinzel.Helpers Namespace](#)

## TextFile Class

Helper class for operations with text files.

### Definition

**Namespace:** [FolkerKinzel.Helpers](#)

**Assembly:** FolkerKinzel.Helpers (in FolkerKinzel.Helpers.dll) Version: 1.0.0+58110188496bad6fb659c326e00a01796f03a8ac

**C#**

```
public static class TextFile
```

**Inheritance**     [Object](#) → TextFile

### Methods

<a href="#">OpenRead</a>	Opens a text file for reading.
<a href="#">OpenWrite</a>	Initializes a new <a href="#">StreamWriter</a> instance.

### See Also

#### Reference

[FolkerKinzel.Helpers Namespace](#)

## TextFile.OpenRead Method

Opens a text file for reading.

### Definition

**Namespace:** [FolkerKinzel.Helpers](#)

**Assembly:** FolkerKinzel.Helpers (in FolkerKinzel.Helpers.dll) Version: 1.0.0+58110188496bad6fb659c326e00a01796f03a8ac

```
C#  
  
public static StreamReader OpenRead(  
    string filePath,  
    Encoding? textEncoding  
)
```

### Parameters

*filePath* [String](#)

The file path.

*textEncoding* [Encoding](#)

The text encoding to be used to read the file, or `null` for [UTF8](#).

### Return Value

[StreamReader](#)

A [StreamReader](#) instance for reading the file.

### Exceptions

<a href="#">ArgumentNullException</a>	<i>filePath</i> is <code>null</code> .
<a href="#">ArgumentException</a>	<i>filePath</i> is not a valid file path.
<a href="#">IOException</a>	I/O error.

### See Also

#### Reference

[TextFile Class](#)

[FolkerKinzel.Helpers Namespace](#)

## TextFile.OpenWrite Method

Initializes a new [StreamWriter](#) instance.

### Definition

**Namespace:** [FolkerKinzel.Helpers](#)

**Assembly:** FolkerKinzel.Helpers (in FolkerKinzel.Helpers.dll) Version: 1.0.0+58110188496bad6fb659c326e00a01796f03a8ac

```
C#
public static StreamWriter OpenWrite(
    string filePath,
    Encoding? textEncoding,
    string? newLine,
    bool append
)
```

### Parameters

*filePath* [String](#)

The file path.

*textEncoding* [Encoding](#)

The text encoding to be used to write the CSV file or `null` for [UTF8](#).

*newLine* [String](#)

The line terminator string to be used, or `null` for [NewLine](#).

*append* [Boolean](#)

If `true`, the file is opened for appending; otherwise, the file will be truncated and overwritten. If the specified file does not exist, this parameter has no effect, and a new file will be created.

### Return Value

[StreamWriter](#)

The newly created [StreamWriter](#) instance.

### Exceptions

<a href="#">ArgumentNullException</a>	<i>filePath</i> is <code>null</code> .
<a href="#">ArgumentException</a>	<i>filePath</i> is not a valid file path.
<a href="#">IOException</a>	I/O-Error

### See Also

#### Reference

[TextFile Class](#)

[FolkerKinzel.Helpers Namespace](#)

## FolkerKinzel.Helpers.Polyfills Namespace

### Polyfills

#### Classes

<a href="#"><u>ArgumentNullException</u></a>	Polyfill for <a href="#"><u>ArgumentNullException</u></a> .
<a href="#"><u>ArgumentOutOfRangeException</u></a>	Polyfill for <a href="#"><u>ArgumentOutOfRangeException</u></a> .



## \_ArgumentNullException Class

Polyfill for [ArgumentNullException](#).

### Definition

**Namespace:** [FolkerKinzel.Helpers.Polyfills](#)

**Assembly:** FolkerKinzel.Helpers (in FolkerKinzel.Helpers.dll) Version: 1.0.0+58110188496bad6fb659c326e00a01796f03a8ac

#### C#

```
public static class _ArgumentNullException
```

**Inheritance**      [Object](#) → [\\_ArgumentNullException](#)

### Methods

#### [ThrowIfNull](#)

Throws an [ArgumentNullException](#) if *argument* is null.

### See Also

#### *Reference*

[FolkerKinzel.Helpers.Polyfills Namespace](#)

## ArgumentNullException.ThrowIfNull Method

Throws an [ArgumentNullException](#) if *argument* is `null`.

### Definition

**Namespace:** [FolkerKinzel.Helpers.Polyfills](#)

**Assembly:** FolkerKinzel.Helpers (in FolkerKinzel.Helpers.dll) Version: 1.0.0+58110188496bad6fb659c326e00a01796f03a8ac

```
C#  
  
public static void ThrowIfNull(  
    Object? argument,  
    string? paramName  
)
```

### Parameters

*argument* [Object](#)

The argument to check.

*paramName* [String](#)

The name of the checked parameter.

### See Also

#### Reference

[ArgumentNullException Class](#)

[FolkerKinzel.Helpers.Polyfills Namespace](#)

## ArgumentOutOfRangeException Class

Polyfill for [ArgumentOutOfRangeException](#).

### Definition

**Namespace:** [FolkerKinzel.Helpers.Polyfills](#)

**Assembly:** FolkerKinzel.Helpers (in FolkerKinzel.Helpers.dll) Version:  
1.0.0+58110188496bad6fb659c326e00a01796f03a8ac

#### C#

```
public static class _ArgumentOutOfRangeException
```

**Inheritance**      [Object](#) → [\\_ArgumentOutOfRangeException](#)

### Methods

<a href="#">ThrowIfNegative</a>	Throws an <a href="#">ArgumentOutOfRangeException</a> if <i>argument</i> is negative.
<a href="#">ThrowIfNegativeOrZero</a>	Throws an <a href="#">ArgumentOutOfRangeException</a> if <i>argument</i> is negative or zero.

See Also

*Reference*

[FolkerKinzel.Helpers.Polyfills Namespace](#)

## ArgumentOutOfRangeException.ThrowIfNegative Method

Throws an [ArgumentOutOfRangeException](#) if *argument* is negative.

### Definition

**Namespace:** [FolkerKinzel.Helpers.Polyfills](#)

**Assembly:** FolkerKinzel.Helpers (in FolkerKinzel.Helpers.dll) Version: 1.0.0+58110188496bad6fb659c326e00a01796f03a8ac

```
C#  
  
public static void ThrowIfNegative(  
    int argument,  
    string? paramName  
)
```

### Parameters

*argument* [Int32](#)

The argument to check.

*paramName* [String](#)

The name of the checked parameter.

### See Also

#### Reference

[ArgumentOutOfRangeException Class](#)

[FolkerKinzel.Helpers.Polyfills Namespace](#)

## ArgumentOutOfRangeException.ThrowIfNegativeOrZero Method

Throws an [ArgumentOutOfRangeException](#) if *argument* is negative or zero.

### Definition

**Namespace:** [FolkerKinzel.Helpers.Polyfills](#)

**Assembly:** FolkerKinzel.Helpers (in FolkerKinzel.Helpers.dll) Version:

1.0.0+58110188496bad6fb659c326e00a01796f03a8ac

```
C#  
  
public static void ThrowIfNegativeOrZero(  
    int argument,  
    string? paramName  
)
```

### Parameters

*argument* [Int32](#)

The argument to check.

*paramName* [String](#)

The name of the checked parameter.

### See Also

#### Reference

[ArgumentOutOfRangeException Class](#)

[FolkerKinzel.Helpers.Polyfills Namespace](#)