

## Table of Contents

### Api Documentation

#### GTiff2Tiles.Core

##### GdalWorker

#### GTiff2Tiles.Core.Constants

##### DateTimePatterns

##### FileExtensions

##### Geodesic

##### Proj

#### GTiff2Tiles.Core.Coordinates

##### Coordinate

##### GeoCoordinate

##### GeodeticCoordinate

##### ICoordinate

##### MercatorCoordinate

##### PixelCoordinate

#### GTiff2Tiles.Core.Enums

##### CoordinateSystem

##### Interpolation

##### TileExtension

#### GTiff2Tiles.Core.Exceptions

##### DirectoryException

##### FileException

##### RasterException

#### GTiff2Tiles.Core.GeoTiffs

##### IGeoTiff

##### Raster

#### GTiff2Tiles.Core.Helpers

##### CheckHelper

##### NetVipsHelper

ProgressHelper

GTiff2Tiles.Core.Images

Area

Band

Size

GTiff2Tiles.Core.Tiles

ITile

Number

RasterTile

Tile

## Namespace GTiff2Tiles.Core

### Classes

#### [GdalWorker](#)

Gdal's methods to work with input files

# Class GdalWorker

Gdal's methods to work with input files

## Inheritance

System.Object  
GdalWorker

## Inherited Members

System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [GTiff2Files.Core](#)

Assembly: GTiff2Files.Core.dll

## Syntax

```
public static class GdalWorker
```

## Fields

### ConvertCoordinateSystemOptions

Options for GdalWarp to convert GeoTiff's coordinate system;

Requires you to add target system param (-t\_srs). Included default args:

```
-overwrite -multi -srcnodata 0  
-of GTiff -ot Byte
```

## Declaration

```
public static readonly string[] ConvertCoordinateSystemOptions
```

## Field Value

Type	Description
System.String[]	

### SrsEpsg3857

-t\_srs EPSG:3857

## Declaration

```
public static readonly string[] SrsEpsg3857
```

## Field Value

Type	Description
System.String[]	

### SrsEpsg4326

-t\_srs EPSG:4326

## Declaration

```
public static readonly string[] SrsEpsg4326
```

## Field Value

Type	Description
System.String[]	

### TempFileName

Name for temporary (converted) GeoTiff

Includes .tif extension, looks like: \_tmp\_converted.tif

## Declaration

```
public static readonly string TempFileName
```

## Field Value

Type	Description
System.String	

## Methods

### ConfigureGdal()

Initialize Gdal, if it hadn't been initialized yet

Declaration

```
public static void ConfigureGdal()
```

ConvertGeoTiffToTargetSystemAsync(String, String, CoordinateSystem, IProgress<Double>)

Converts current GeoTiff to a new GeoTiff with target [CoordinateSystem](#) through GdalWarp

Declaration

```
public static Task ConvertGeoTiffToTargetSystemAsync(string inputFilePath, string outputFilePath, CoordinateSystem targetSystem, IProgress<double> progress = null)
```

Parameters

Type	Name	Description
System.String	inputFilePath	Input GeoTiff's path
System.String	outputFilePath	Output GeoTiff's path
<a href="#">CoordinateSystem</a>	targetSystem	Target <a href="#">CoordinateSystem</a>
System.IProgress<System.Double> progress		GdalWarp's progress null by default

Returns

Type	Description
System.Threading.Tasks.Task	

Exceptions

Type	Condition
System.NotSupportedException	

GetCoordinateSystem(String)

Gets supported coordinate system from proj string of GeoTiff

Declaration

```
public static CoordinateSystem GetCoordinateSystem(string projString)
```

Parameters

Type	Name	Description
System.String	projString	Proj string of input GeoTiff

Returns

Type	Description
<a href="#">CoordinateSystem</a>	Input file's <a href="#">CoordinateSystem</a>

Exceptions

Type	Condition
System.ArgumentNullException	

GetGeoTransform(String)

Gets the coordinates and pixel sizes of image

Declaration

```
public static double[] GetGeoTransform(string inputFilePath)
```

Parameters

Type	Name	Description
System.String	inputFilePath	Input GeoTiff's path

Returns

Type	Description
System.Double[]	Array of double coordinates and pixel sizes if everything is OK; null otherwise

GetImageBorders(String, Size, CoordinateSystem)

Gets the coordinates borders of the input Geotiff file

Declaration

```
public static (GeoCoordinate minCoordinate, GeoCoordinate maxCoordinate) GetImageBorders(string inputFilePath, Size size, CoordinateSystem coordinateSystem)
```

Parameters

Type	Name	Description
System.String	inputFilePath	Input GeoTiff's path
<a href="#">Size</a>	size	Image's <a href="#">Sizes</a>
<a href="#">CoordinateSystem</a>	coordinateSystem	Image's coordinate system

Returns

Type	Description
System.ValueTuple< <a href="#">GeoCoordinate</a> , <a href="#">GeoCoordinate</a> >	System.ValueTuple'2 of <a href="#">GeoCoordinates</a> of image's borders

Exceptions

Type	Condition
System.ArgumentNullException	
System.NotSupportedException	

GetProjString(String)

Gets proj System.String of input file

Declaration

```
public static string GetProjString(string inputFilePath)
```

Parameters

Type	Name	Description
System.String	inputFilePath	Input GeoTiff's path

Returns

Type	Description
System.String	Proj System.String if everything OK; null otherwise

GetProjStringAsync(String)

Gets proj System.String of input file

Declaration

```
public static Task<string> GetProjStringAsync(string inputFilePath)
```

Parameters

Type	Name	Description
System.String	inputFilePath	Input GeoTiff's path

Returns

Type	Description
System.Threading.Tasks.Task<System.String>	Proj System.String if everything OK; null otherwise

InfoAsync(String, String[])

Runs GdalInfo with passed parameters

Declaration

```
public static Task<string> InfoAsync(string inputFilePath, string[] options = null)
```

Parameters

Type	Name	Description
System.String	inputFilePath	Input GeoTiff's path

Type	Name	Description
System.String[] options		Array of string parameters for GdalInfo
		null by default

Returns

Type	Description
System.Threading.Tasks.Task<System.String>	System.String from GdalInfo if everything OK; null otherwise

WarpAsync(String, String, String[], IProgress<Double>)

Runs GdalWarp with passed parameters

Declaration

```
public static Task WarpAsync(string inputFilePath, string outputFilePath, string[] options, IProgress<double> progress = null)
```

Parameters

Type	Name	Description
System.String	inputFilePath	Input GeoTiff's path
System.String	outputFilePath	Output file's path
System.String[] options		Array of string parameters
		See <a href="#">ConvertCoordinateSystemOptions</a> field for more info
System.IProgress<System.Double> progress		GdalWarp's progress
		null by default

Returns

Type	Description
System.Threading.Tasks.Task	

Exceptions

Type	Condition
System.ArgumentNullException	

# Namespace GTiff2Tiles.Core.Constants

## Classes

### [Date Time Patterns](#)

String patterns for System.DateTime

### [File Extensions](#)

Used extensions

### [Geodesic](#)

Some geo-related constants

### [Proj](#)

Proj constants



# Class DateTimePatterns

String patterns for System.DateTime

## Inheritance

System.Object  
DateTimePatterns

## Inherited Members

System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [GTiff2Tiles.Core.Constants](#)

Assembly: GTiff2Tiles.Core.dll

## Syntax

```
public static class DateTimePatterns
```

## Fields

### LongWithMs

Year, month, date, hour, minutes, seconds, ms

#### Declaration

```
public const string LongWithMs = "yyyyMMddHHmmssfff"
```

#### Field Value

Type	Description
System.String	

### ShortToDate

Year, month and date

#### Declaration

```
public const string ShortToDate = "yyyyMMdd"
```

#### Field Value

Type	Description
System.String	

### ShortToMonth

Year and month

#### Declaration

```
public const string ShortToMonth = "yyyyMM"
```

#### Field Value

Type	Description
System.String	

# Class FileExtensions

Used extensions

**Inheritance**

System.Object  
FileExtensions

**Inherited Members**

System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [G:Tiff2Tiles.Core.Constants](#)

Assembly: G:Tiff2Tiles.Core.dll

**Syntax**

```
public static class FileExtensions
```

## Fields

### Jpg

.jpg

**Declaration**

```
public const string Jpg = ".jpg"
```

**Field Value**

Type	Description
System.String	

### Png

.png

**Declaration**

```
public const string Png = ".png"
```

**Field Value**

Type	Description
System.String	

### Tif

.tif

**Declaration**

```
public const string Tif = ".tif"
```

**Field Value**

Type	Description
System.String	

## Webp

.webp

### Declaration

```
public const string Webp = ".webp"
```

### Field Value

Type	Description
System.String	

# Class Geodesic

Some geo-related constants

## Inheritance

System.Object  
Geodesic

## Inherited Members

System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [G TIFF2 Tiles.Core.Constants](#)

Assembly: G TIFF2 Tiles.Core.dll

## Syntax

```
public static class Geodesic
```

## Fields

### EquatorRadius

Radius of Earth, measured at equator

#### Declaration

```
public const double EquatorRadius = 6378137
```

#### Field Value

Type	Description
System.Double	

### OriginShift

Approximately 20037508.342789244

#### Declaration

```
public const double OriginShift = 20037508.342789244
```

#### Field Value

Type	Description
System.Double	

### PolarRadius

Radius of Earth, measured at pole

#### Declaration

```
public const double PolarRadius = 6356752.314245
```

#### Field Value

Type	Description
System.Double	

# Class Proj

Proj constants

## Inheritance

System.Object  
Proj

## Inherited Members

System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [GTiff2Tiles.Core.Constants](#)

Assembly: GTiff2Tiles.Core.dll

## Syntax

```
public static class Proj
```

## Fields

### DatumWgs84

For EPSG:4326 AND World Mercator, BUT not Spherical Mercator (EPSG:3857)

#### Declaration

```
public const string DatumWgs84 = "+datum=WGS84"
```

#### Field Value

Type	Description
System.String	

### LongLatFull

Full PROJ string for LongLat (EPSG:4326) projection

#### Declaration

```
public static readonly string LongLatFull
```

#### Field Value

Type	Description
System.String	

### MercFull

Full PROJ string for Spherical Mercator (EPSG:3857) projection

#### Declaration

```
public static readonly string MercFull
```

#### Field Value

Type	Description
System.String	

## NoDefs

+no\_defs

### Declaration

```
public const string NoDefs = "+no_defs"
```

### Field Value

Type	Description
System.String	

## ProjLongLat

+proj=longlat

### Declaration

```
public const string ProjLongLat = "+proj=longlat"
```

### Field Value

Type	Description
System.String	

## ProjMerc

+proj=merc

### Declaration

```
public const string ProjMerc = "+proj=merc"
```

### Field Value

Type	Description
System.String	



# Namespace GTiff2Tiles.Core.Coordinates

## Classes

### [Coordinate](#)

Basic implementation of [ICoordinate](#) interface

### [GeoCoordinate](#)

Class for geographical coordinates

### [GeodeticCoordinate](#)

Class for EPSG:4326 coordinates

### [MercatorCoordinate](#)

Class for EPSG:3857 coordinates

### [PixelCoordinate](#)

Coordinates in pixels

## Interfaces

### [ICoordinate](#)

Interface for any coordinate

# Class Coordinate

Basic implementation of [ICoordinate](#) interface

## Inheritance

System.Object  
Coordinate  
[GeoCoordinate](#)  
[PixelCoordinate](#)

## Implements

[ICoordinate](#)  
System.IEquatable<[ICoordinate](#)>

## Inherited Members

System.Object.Equals(System.Object, System.Object)  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [GTiff2Tiles.Core.Coordinates](#)

Assembly: GTiff2Tiles.Core.dll

## Syntax

```
public class Coordinate : ICoordinate, IEquatable<ICoordinate>
```

## Constructors

### Coordinate(Double, Double)

Create instance of class

## Declaration

```
protected Coordinate(double x, double y)
```

## Parameters

Type	Name	Description
System.Double	x	X coordinate value
System.Double	y	Y coordinate value

## Properties

### X

X coordinate value or Longitude

## Declaration

```
public virtual double X { get; }
```

## Property Value

Type	Description
------	-------------

Type	Description
System.Double	

## Y

Y coordinate value or Latitude

### Declaration

```
public virtual double Y { get; }
```

### Property Value

Type	Description
System.Double	

## Methods

### Add(Coordinate)

Sum coordinates

### Declaration

```
public Coordinate Add(Coordinate other)
```

### Parameters

Type	Name	Description
<a href="#">Coordinate</a>	other	<a href="#">Coordinate</a> to add

### Returns

Type	Description
<a href="#">Coordinate</a>	New coordinate

### DegreesToRadians(Double)

Converts degrees to radians

### Declaration

```
public static double DegreesToRadians(double degrees)
```

### Parameters

Type	Name	Description
System.Double	degrees	Value to convert

### Returns

Type	Description
System.Double	Converted radians

### Divide(Coordinate)

Divide coordinates

Declaration

```
public Coordinate Divide(Coordinate other)
```

Parameters

Type	Name	Description
<a href="#">Coordinate</a>	other	<a href="#">Coordinate</a> to divide on

Returns

Type	Description
<a href="#">Coordinate</a>	New coordinate

Equals(ICoordinate)

Declaration

```
public bool Equals(ICoordinate other)
```

Parameters

Type	Name	Description
<a href="#">ICoordinate</a>	other	

Returns

Type	Description
System.Boolean	

Equals(Object)

Declaration

```
public override bool Equals(object coordinate)
```

Parameters

Type	Name	Description
System.Object	coordinate	

Returns

Type	Description
System.Boolean	

Overrides

```
System.Object.Equals(System.Object)
```

GetHashCode()

Declaration

```
public override int GetHashCode()
```

Returns

Type	Description
System.Int32	

Overrides

System.Object.GetHashCode()

Multiply(Coordinate)

Multiply coordinates

Declaration

```
public Coordinate Multiply(Coordinate other)
```

Parameters

Type	Name	Description
<a href="#">Coordinate</a>	other	<a href="#">Coordinate</a> to multiply

Returns

Type	Description
<a href="#">Coordinate</a>	New coordinate

RadiansToDegrees(Double)

Converts radians to degrees

Declaration

```
public static double RadiansToDegrees(double radians)
```

Parameters

Type	Name	Description
System.Double	radians	Value to convert

Returns

Type	Description
System.Double	Converted degrees

Round<T>(T, Int32)

Round coordinate's [X](#) and [Y](#)

Declaration

```
public static T Round<T>(T coordinate, int digits)
    where T : ICoordinate
```

Parameters

Type	Name	Description
------	------	-------------

Type	Name	Description
T	coordinate	<a href="#">ICoordinate</a> to round
		System.Int32 digits

Returns

Type	Description
T	Rounded coordinate

Type Parameters

Name	Description
T	Child of <a href="#">ICoordinate</a>

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentOutOfRangeException	

**Round<T>(Int32)**

Round coordinate's [X](#) and [Y](#)

Declaration

```
public T Round<T>(int digits)
    where T : ICoordinate
```

Parameters

Type	Name	Description
		Number of digits after zero in return false
System.Int32 digits		Must be bigger or equal, than 0

Returns

Type	Description
T	Rounded coordinate

Type Parameters

Name	Description
T	Child of <a href="#">ICoordinate</a>

**Subtract(Coordinate)**

Subtruct coordinates

Declaration

```
public Coordinate Subtract(Coordinate other)
```

Parameters

Type	Name	Description
<a href="#">Coordinate</a>	other	<a href="#">Coordinate</a> to subtract

Returns

Type	Description
<a href="#">Coordinate</a>	New coordinate

ToNumber(Int32, Size, Boolean)

Calculate [Number](#) for current [ICoordinate](#)

Declaration

```
public virtual Number ToNumber(int z, Size tileSize, bool tmsCompatible)
```

Parameters

Type	Name	Description
System.Int32	z	Zoom
		Must be >= 0
<a href="#">Size</a>	tileSize	<a href="#">ITile</a> 's size
System.Boolean tmsCompatible	Is <a href="#">ITile</a>	tms compatible?

Returns

Type	Description
<a href="#">Number</a>	<a href="#">Number</a> in which this <a href="#">ICoordinate</a> belongs

Operators

Addition(Coordinate, Coordinate)

Sum coordinates

Declaration

```
public static Coordinate operator +(Coordinate coordinate1, Coordinate coordinate2)
```

Parameters

Type	Name	Description
<a href="#">Coordinate</a>	coordinate1	Coordinate 1
<a href="#">Coordinate</a>	coordinate2	Coordinate 2

Returns

Type	Description
<a href="#">Coordinate</a>	New coordinate

**Division(Coordinate, Coordinate)**

Divide coordinates

Declaration

```
public static Coordinate operator /(Coordinate coordinate1, Coordinate coordinate2)
```

Parameters

Type	Name	Description
<a href="#">Coordinate</a>	coordinate1	Coordinate 1
<a href="#">Coordinate</a>	coordinate2	Coordinate 2

Returns

Type	Description
<a href="#">Coordinate</a>	New coordinate

**Equality(Coordinate, Coordinate)**

Check two coordinates for equality

Declaration

```
public static bool operator ==(Coordinate coordinate1, Coordinate coordinate2)
```

Parameters

Type	Name	Description
<a href="#">Coordinate</a>	coordinate1	Coordinate 1
<a href="#">Coordinate</a>	coordinate2	Coordinate 2

Returns

Type	Description
System.Boolean	true if coordinates are equal; falseotherwise

**Inequality(Coordinate, Coordinate)**

Check two coordinates for non-equality

Declaration

```
public static bool operator !=(Coordinate coordinate1, Coordinate coordinate2)
```



Parameters

Type	Name	Description
<a href="#">Coordinate</a>	coordinate1	Coordinate 1
<a href="#">Coordinate</a>	coordinate2	Coordinate 2

Returns

Type	Description
System.Boolean	true if coordinates are not equal; falseotherwise

Multiply([Coordinate](#), [Coordinate](#))

Multiply coordinates

Declaration

```
public static Coordinate operator *(Coordinate coordinate1, Coordinate coordinate2)
```

Parameters

Type	Name	Description
<a href="#">Coordinate</a>	coordinate1	Coordinate 1
<a href="#">Coordinate</a>	coordinate2	Coordinate 2

Returns

Type	Description
<a href="#">Coordinate</a>	New coordinate

Subtraction([Coordinate](#), [Coordinate](#))

Subtruct coordinates

Declaration

```
public static Coordinate operator -(Coordinate coordinate1, Coordinate coordinate2)
```

Parameters

Type	Name	Description
<a href="#">Coordinate</a>	coordinate1	Coordinate 1
<a href="#">Coordinate</a>	coordinate2	Coordinate 2

Returns

Type	Description
------	-------------

Type	Description
<a href="#">Coordinate</a>	New coordinate

## Implements

[ICoordinate](#)  
System.IEquatable<T>

# Class GeoCoordinate

Class for geographical coordinates

Inheritance

System.Object  
[Coordinate](#)  
GeoCoordinate  
[GeodeticCoordinate](#)  
[MercatorCoordinate](#)

Implements

[ICoordinate](#)  
System.IEquatable<[ICoordinate](#)>

Inherited Members

[Coordinate.X](#)  
[Coordinate.Y](#)  
[Coordinate.DegreesToRadians\(Double\)](#)  
[Coordinate.RadiansToDegrees\(Double\)](#)  
[Coordinate.Round<T>\(Int32\)](#)  
[Coordinate.Round<T>\(T, Int32\)](#)  
[Coordinate.Equals\(Object\)](#)  
[Coordinate.GetHashCode\(\)](#)  
[Coordinate.Equals\(ICoordinate\)](#)  
[Coordinate.Add\(Coordinate\)](#)  
[Coordinate.Subtract\(Coordinate\)](#)  
[Coordinate.Multiply\(Coordinate\)](#)  
[Coordinate.Divide\(Coordinate\)](#)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [GTile2Tiles.Core.Coordinates](#)

Assembly: GTile2Tiles.Core.dll

Syntax

```
public class GeoCoordinate : Coordinate, ICoordinate, IEquatable<ICoordinate>
```

## Constructors

### GeoCoordinate(Double, Double)

Create instance of class

Declaration

```
protected GeoCoordinate(double x, double y)
```

Parameters

Type	Name	Description
System.Double	x	X coordinate value
System.Double	y	Y coordinate value

## Methods

### GetNumbers(GeoCoordinate, GeoCoordinate, Int32, Size, Boolean)

Gets [Numbers](#) for specified [GeoCoordinates](#)

Declaration

```
public static (Number minNumber, Number maxNumber) GetNumbers(GeoCoordinate minCoordinate, GeoCoordinate maxCoordinate, int z, Size tileSize, bool tmsCompatible)
```

Parameters

Type	Name	Description
<a href="#">GeoCoordinate</a>	minCoordinate	Minimal <a href="#">GeoCoordinate</a>
<a href="#">GeoCoordinate</a>	maxCoordinate	Maximal <a href="#">GeoCoordinate</a>
System.Int32	z	Zoom
<a href="#">Size</a>	tileSize	<a href="#">ITile</a> 's size Must be square

Type	Name	Description
System.Boolean	tmsCompatible	Is <a href="#">ITile</a> tms compatible?

Returns

Type	Description
System.ValueTuple< <a href="#">Number</a> , <a href="#">Number</a> >	System.ValueTuple`2 of <a href="#">Numbers</a>

Resolution(Int32, Size, CoordinateSystem)

Resolution for given zoom level (measured at Equator)

Declaration

```
public static double Resolution(int z, Size tileSize, CoordinateSystem coordinateSystem)
```

Parameters

Type	Name	Description
System.Int32	z	Zoom Must be >= 0
<a href="#">Size</a>	tileSize	<a href="#">ITile</a> 's size Must be square

[CoordinateSystem](#) coordinateSystem Coordinate system

Returns

Type	Description
System.Double	Resolution value or -1.0 if something goes wrong

Exceptions

Type	Condition
System.NotSupportedException	

ToNumber(Int32, Size, Boolean)

Calculate [Number](#) for current [ICoordinate](#)

Declaration

```
public override Number ToNumber(int z, Size tileSize, bool tmsCompatible)
```

Parameters

Type	Name	Description
System.Int32	z	Zoom Must be >= 0
<a href="#">Size</a>	tileSize	<a href="#">ITile</a> 's size

System.Boolean tmsCompatible Is [ITile](#) tms compatible?

Returns

Type	Description
<a href="#">Number</a> <a href="#">Number</a>	in which this <a href="#">ICoordinate</a> belongs

Overrides

[Coordinate.ToNumber\(Int32, Size, Boolean\)](#)

Exceptions

Type	Condition
System.ArgumentOutOfRangeException	

ToPixelCoordinate(Int32, Size)

Convert current [GeoCoordinate](#) to [PixelCoordinate](#)

Declaration

```
public virtual PixelCoordinate ToPixelCoordinate(int z, Size tileSize)
```

Parameters

Type	Name	Description
System.Int32 z		Zoom
		Must be >= 0
<a href="#">Size</a>	tileSize	<a href="#">ITile</a> 's size
		Must be square

Returns

Type	Description
<a href="#">PixelCoordinate</a>	Converted <a href="#">PixelCoordinate</a>

ZoomForPixelSize(Int32, Size, CoordinateSystem, Int32, Int32)

Calculate zoom from known pixel size

Declaration

```
public static int ZoomForPixelSize(int pixelSize, Size tileSize, CoordinateSystem coordinateSystem, int minZ = 0, int maxZ = 32)
```

Parameters

Type	Name	Description
System.Int32	pixelSize	Pixel size
<a href="#">Size</a>	tileSize	<a href="#">ITile</a> 's size
		Must be square
<a href="#">CoordinateSystem</a>	coordinateSystem	Coordinate system
System.Int32	minZ	Minimal zoom
		Must be >= 0 and lesser or equal, than <code>maxZ</code>
		0 by default
System.Int32	maxZ	Maximal zoom
		Must be >= 0 and bigger or equal, than <code>minZ</code>
		32 by default

Returns

Type	Description
System.Int32	Approximate zoom value

Implements

[ICoordinate](#)  
System.IEquatable<T>

# Class GeodeticCoordinate

Class for EPSG:4326 coordinates

## Inheritance

System.Object  
[Coordinate](#)  
[GeoCoordinate](#)  
GeodeticCoordinate

## Implements

[ICoordinate](#)  
System.IEquatable<[ICoordinate](#)>

## Inherited Members

[GeoCoordinate.ToNumber\(Int32, Size, Boolean\)](#)  
[GeoCoordinate.GetNumbers\(GeoCoordinate, GeoCoordinate, Int32, Size, Boolean\)](#)  
[GeoCoordinate.Resolution\(Int32, Size, CoordinateSystem\)](#)  
[GeoCoordinate.ZoomForPixelSize\(Int32, Size, CoordinateSystem, Int32, Int32\)](#)  
[Coordinate.X](#)  
[Coordinate.Y](#)  
[Coordinate.DegreesToRadians\(Double\)](#)  
[Coordinate.RadiansToDegrees\(Double\)](#)  
[Coordinate.Round<T>\(Int32\)](#)  
[Coordinate.Round<T>\(T, Int32\)](#)  
[Coordinate.Equals\(Object\)](#)  
[Coordinate.GetHashCode\(\)](#)  
[Coordinate.Equals\(ICoordinate\)](#)  
[Coordinate.Add\(Coordinate\)](#)  
[Coordinate.Subtract\(Coordinate\)](#)  
[Coordinate.Multiply\(Coordinate\)](#)  
[Coordinate.Divide\(Coordinate\)](#)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [GTiff2Tiles.Core.Coordinates](#)

Assembly: GTiff2Tiles.Core.dll

## Syntax

```
public class GeodeticCoordinate : GeoCoordinate, ICoordinate, IEquatable<ICoordinate>
```

## Constructors

### GeodeticCoordinate(Double, Double)

#### Declaration

```
public GeodeticCoordinate(double longitude, double latitude)
```

#### Parameters

Type	Name	Description
		<a href="#">X</a> or Longitude
System.Double	longitude	Must be in range [-180.0, 180.0]

Type	Name	Description
System.Double	latitude	<a href="#">Y</a> or Latitude Must be in range [-90.0, 90.0]

#### Exceptions

Type	Condition
System.ArgumentOutOfRangeException	

## Fields

### MaxPossibleLatValue

Maximal possible value of latitude for EPSG:4326

#### Declaration

```
public const double MaxPossibleLatValue = 90
```

#### Field Value

Type	Description
System.Double	

### MaxPossibleLonValue

Maximal possible value of longitude for EPSG:4326

#### Declaration

```
public const double MaxPossibleLonValue = 180
```

#### Field Value

Type	Description
System.Double	

### MinPossibleLatValue

Minimal possible value of latitude for EPSG:4326

#### Declaration

```
public const double MinPossibleLatValue = -90
```

#### Field Value

Type	Description
System.Double	

### MinPossibleLonValue

Minimal possible value of longitude for EPSG:4326

#### Declaration

```
public const double MinPossibleLonValue = -180
```

#### Field Value

Type	Description
System.Double	

## Properties

### Latitude

Analogue of [Y](#)

#### Declaration

```
public double Latitude { get; }
```

#### Property Value

Type	Description
System.Double	

### Longitude

Analogue of [X](#)

#### Declaration

```
public double Longitude { get; }
```

#### Property Value

Type	Description
System.Double	

## Methods

### Resolution(Int32, Size)

Resolution for given zoom level (measured at Equator)

#### Declaration

```
public static double Resolution(int z, Size tileSize)
```

#### Parameters

Type	Name	Description
System.Int32	z	
<a href="#">Size</a>	tileSize	

#### Returns

Type	Description
System.Double	Resolution value or -1.0 if something goes wrong

#### Exceptions

Type	Condition
System.ArgumentOutOfRangeException	
System.ArgumentNullException	
System.ArgumentException	



**ToMercatorCoordinate()**

Convert current coordinate to [MercatorCoordinate](#)

**Declaration**

```
public MercatorCoordinate ToMercatorCoordinate()
```

**Returns**

Type	Description
<a href="#">MercatorCoordinate</a>	Converted <a href="#">MercatorCoordinate</a>

**ToPixelCoordinate(Int32, Size)**

Convert current [GeoCoordinate](#) to [PixelCoordinate](#)

**Declaration**

```
public override PixelCoordinate ToPixelCoordinate(int z, Size tileSize)
```

**Parameters**

Type	Name	Description
System.Int32	z	Zoom
		Must be >= 0
<a href="#">Size</a>	tileSize	<a href="#">ITile</a> 's size Must be square

**Returns**

Type	Description
<a href="#">PixelCoordinate</a>	Converted <a href="#">PixelCoordinate</a>

**Overrides**

[GeoCoordinate.ToPixelCoordinate\(Int32, Size\)](#)

**Exceptions**

Type	Condition
System.ArgumentOutOfRangeException	

**Implements**

[ICoordinate](#)  
System.IEquatable<T>

# Interface ICoordinate

Interface for any coordinate

## Inherited Members

System.IEquatable<GTiff2Tiles.Core.Coordinates.ICoordinate>.Equals(GTiff2Tiles.Core.Coordinates.ICoordinate)

Namespace: [GTiff2Tiles.Core.Coordinates](#)

Assembly: GTiff2Tiles.Core.dll

## Syntax

```
public interface ICoordinate : IEquatable<ICoordinate>
```

## Properties

### X

X coordinate value or Longitude

#### Declaration

```
double X { get; }
```

#### Property Value

Type	Description
System.Double	

### Y

Y coordinate value or Latitude

#### Declaration

```
double Y { get; }
```

#### Property Value

Type	Description
System.Double	

## Methods

### Round<T>(Int32)

Round coordinate's [X](#) and [Y](#)

#### Declaration

```
T Round<T>(int digits)
    where T : ICoordinate
```

#### Parameters

Type	Name	Description
System.Int32	digits	Number of digits after zero in return false Must be bigger or equal, than 0

Returns

Type	Description
T	Rounded coordinate

Type Parameters

Name	Description
T	Child of <a href="#">ICoordinate</a>

ToNumber(Int32, Size, Boolean)

Calculate [Number](#) for current [ICoordinate](#)

Declaration

Number ToNumber(int z, Size tileSize, bool tmsCompatible)

Parameters

Type	Name	Description
System.Int32	z	Zoom
		Must be >= 0
<a href="#">Size</a>	tileSize	<a href="#">ITile</a> 's size
System.Boolean tmsCompatible	Is <a href="#">ITile</a> tms compatible?	

Returns

Type	Description
<a href="#">Number</a> <a href="#">Number</a>	in which this <a href="#">ICoordinate</a> belongs

# Class MercatorCoordinate

Class for EPSG:3857 coordinates

## Inheritance

System.Object  
[Coordinate](#)  
[GeoCoordinate](#)  
MercatorCoordinate

## Implements

[ICoordinate](#)  
System.IEquatable<[ICoordinate](#)>

## Inherited Members

[GeoCoordinate.ToNumber\(Int32, Size, Boolean\)](#)  
[GeoCoordinate.GetNumbers\(GeoCoordinate, GeoCoordinate, Int32, Size, Boolean\)](#)  
[GeoCoordinate.Resolution\(Int32, Size, CoordinateSystem\)](#)  
[GeoCoordinate.ZoomForPixelSize\(Int32, Size, CoordinateSystem, Int32, Int32\)](#)  
[Coordinate.X](#)  
[Coordinate.Y](#)  
[Coordinate.DegreesToRadians\(Double\)](#)  
[Coordinate.RadiansToDegrees\(Double\)](#)  
[Coordinate.Round<T>\(Int32\)](#)  
[Coordinate.Round<T>\(T, Int32\)](#)  
[Coordinate.Equals\(Object\)](#)  
[Coordinate.GetHashCode\(\)](#)  
[Coordinate.Equals\(ICoordinate\)](#)  
[Coordinate.Add\(Coordinate\)](#)  
[Coordinate.Subtract\(Coordinate\)](#)  
[Coordinate.Multiply\(Coordinate\)](#)  
[Coordinate.Divide\(Coordinate\)](#)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [GTiff2Tiles.Core.Coordinates](#)

Assembly: GTiff2Tiles.Core.dll

## Syntax

```
public class MercatorCoordinate : GeoCoordinate, ICoordinate, IEquatable<ICoordinate>
```

## Constructors

### MercatorCoordinate(Double, Double)

#### Declaration

```
public MercatorCoordinate(double longitude, double latitude)
```

#### Parameters

Type	Name	Description
		<a href="#">X</a> or Longitude
System.Double	longitude	Must be in range [-20026376.39, 20026376.39]

Type	Name	Description
System.Double	latitude	<a href="#">Y</a> or Latitude Must be in range [-20048966.10, 20048966.10]

#### Exceptions

Type	Condition
System.ArgumentOutOfRangeException	

## Fields

### MaxPossibleLatValue

Maximal possible value of latitude for EPSG:3857

#### Declaration

```
public const double MaxPossibleLatValue = 20048966.1
```

#### Field Value

Type	Description
System.Double	

### MaxPossibleLonValue

Maximal possible value of longitude for EPSG:3857

#### Declaration

```
public const double MaxPossibleLonValue = 20037508.35
```

#### Field Value

Type	Description
System.Double	

### MinPossibleLatValue

Minimal possible value of latitude for EPSG:3857

#### Declaration

```
public const double MinPossibleLatValue = -20048966.1
```

#### Field Value

Type	Description
System.Double	

### MinPossibleLonValue

Minimal possible value of longitude for EPSG:3857

#### Declaration

```
public const double MinPossibleLonValue = -20037508.35
```

#### Field Value

Type	Description
System.Double	

## Methods

### Resolution(Int32, Size)

Resolution for given zoom level (measured at Equator)

#### Declaration

```
public static double Resolution(int z, Size tileSize)
```

#### Parameters

Type	Name	Description
System.Int32	z	
<a href="#">Size</a>	tileSize	

#### Returns

Type	Description
System.Double	Resolution value or -1.0 if something goes wrong

#### Exceptions

Type	Condition
System.ArgumentOutOfRangeException	
System.ArgumentNullException	
System.ArgumentException	

### ToGeodeticCoordinate()

Convert current coordinate to [GeodeticCoordinate](#)

#### Declaration

```
public GeodeticCoordinate ToGeodeticCoordinate()
```

#### Returns

Type	Description
<a href="#">GeodeticCoordinate</a>	Converted <a href="#">GeodeticCoordinate</a>

### ToPixelCoordinate(Int32, Size)

Convert current [GeoCoordinate](#) to [PixelCoordinate](#)

#### Declaration

```
public override PixelCoordinate ToPixelCoordinate(int z, Size tileSize)
```

#### Parameters

Type	Name	Description
------	------	-------------

Type	Name	Description
System.Int32	z	Zoom Must be $\geq 0$

<a href="#">Size</a>	tileSize	<a href="#">ITile</a> 's size Must be square
----------------------	----------	---

#### Returns

Type	Description
<a href="#">PixelCoordinate</a>	Converted <a href="#">PixelCoordinate</a>

#### Overrides

[GeoCoordinate.ToPixelCoordinate\(Int32, Size\)](#)

#### Implements

[ICoordinate](#)  
System.IEquatable<T>

# Class PixelCoordinate

Coordinates in pixels

Inheritance

System.Object  
[Coordinate](#)  
PixelCoordinate

Implements

[ICoordinate](#)  
System.IEquatable<[ICoordinate](#)>

Inherited Members

[Coordinate.X](#)  
[Coordinate.Y](#)  
[Coordinate.DegreesToRadians\(Double\)](#)  
[Coordinate.RadiansToDegrees\(Double\)](#)  
[Coordinate.Round<T>\(Int32\)](#)  
[Coordinate.Round<T>\(T, Int32\)](#)  
[Coordinate.Equals\(Object\)](#)  
[Coordinate.GetHashCode\(\)](#)  
[Coordinate.Equals\(ICoordinate\)](#)  
[Coordinate.Add\(Coordinate\)](#)  
[Coordinate.Subtract\(Coordinate\)](#)  
[Coordinate.Multiply\(Coordinate\)](#)  
[Coordinate.Divide\(Coordinate\)](#)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [G TIFF2 Tiles.Core.Coordinates](#)

Assembly: [G TIFF2 Tiles.Core.dll](#)

Syntax

```
public class PixelCoordinate : Coordinate, ICoordinate, IEquatable<ICoordinate>
```

## Constructors

### PixelCoordinate(Double, Double)

Create instance of class

Declaration

```
public PixelCoordinate(double x, double y)
```

Parameters

Type	Name	Description
System.Double x		X coordinate value
		Must be >= 0
System.Double y		Y coordinate value
		Must be >= 0

Exceptions

Type	Condition
System.ArgumentOutOfRangeException	

## Methods

### ToGeoCoordinate(CoordinateSystem, CoordinateSystem, Int32, Size)

Convert current coordinate to child of [GeoCoordinate](#)

Declaration

```
public GeoCoordinate ToGeoCoordinate(CoordinateSystem inputCoordinateSystem, CoordinateSystem targetCoordinateSystem, int z, Size tileSize)
```



Parameters

Type	Name	Description
<a href="#">CoordinateSystem</a>	inputCoordinateSystem	<a href="#">CoordinateSystem</a> from which pixel coordinates were maid
<a href="#">CoordinateSystem</a>	targetCoordinateSystem	Coordinate system
System.Int32	z	Zoom Must be >= 0
<a href="#">Size</a>	tileSize	<a href="#">ITile</a> 's size Must be square

Returns

Type	Description
<a href="#">GeoCoordinate</a>	Converted to <a href="#">GeoCoordinate</a> value or null if something goes wrong

Exceptions

Type	Condition
System.NotSupportedException	

ToGeodeticCoordinate(CoordinateSystem, Int32, Size)

Convert current coordinate to [GeodeticCoordinate](#)

Declaration

```
public GeodeticCoordinate ToGeodeticCoordinate(CoordinateSystem inputCoordinateSystem, int z, Size tileSize)
```

Parameters

Type	Name	Description
<a href="#">CoordinateSystem</a>	inputCoordinateSystem	<a href="#">CoordinateSystem</a> from which pixel coordinates were maid
System.Int32	z	Zoom Must be >= 0
<a href="#">Size</a>	tileSize	<a href="#">ITile</a> 's size Must be square

Returns

Type	Description
<a href="#">GeodeticCoordinate</a>	Converted <a href="#">GeodeticCoordinate</a>

Exceptions

Type	Condition
System.ArgumentOutOfRangeException	
System.NotSupportedException	

ToMercatorCoordinate(CoordinateSystem, Int32, Size)

Convert current coordinate to [MercatorCoordinate](#)

Declaration

```
public MercatorCoordinate ToMercatorCoordinate(CoordinateSystem inputCoordinateSystem, int z, Size tileSize)
```

Parameters

Type	Name	Description
------	------	-------------

Type	Name	Description
<a href="#">CoordinateSystem</a>	inputCoordinateSystem	<a href="#">CoordinateSystem</a> from which pixel coordinates were maid
System.Int32	z	Zoom Must be >= 0
<a href="#">Size</a>	tileSize	<a href="#">ITile</a> 's size Must be square

#### Returns

Type	Description
<a href="#">MercatorCoordinate</a>	Converted <a href="#">MercatorCoordinate</a>

#### Exceptions

Type	Condition
System.ArgumentOutOfRangeException	
System.NotSupportedException	

#### ToNumber(Int32, Size, Boolean)

Calculate [Number](#) for current [ICoordinate](#)

#### Declaration

```
public override Number ToNumber(int z, Size tileSize, bool tmsCompatible)
```

#### Parameters

Type	Name	Description
System.Int32	z	
<a href="#">Size</a>	tileSize	<a href="#">ITile</a> 's size Must be square

System.Boolean tmsCompatible

#### Returns

Type	Description
<a href="#">Number</a>	<a href="#">Number</a> in which this <a href="#">ICoordinate</a> belongs

#### Overrides

[Coordinate.ToNumber\(Int32, Size, Boolean\)](#)

#### Exceptions

Type	Condition
System.ArgumentOutOfRangeException	
System.ArgumentNullException	
System.ArgumentException	

#### ToRasterPixelCoordinate(Int32, Size)

Move the origin of pixel coordinates to top-left corner

#### Declaration

```
public PixelCoordinate ToRasterPixelCoordinate(int z, Size tileSize)
```

#### Parameters

Type	Name	Description
------	------	-------------

Type	Name	Description
System.Int32 z		Zoom
		Must be $\geq 0$

<a href="#">Size</a>		<a href="#">ITile</a> 's size
	tileSize	Must be square

Returns

Type	Description
<a href="#">PixelCoordinate</a>	Converted <a href="#">PixelCoordinate</a>

Exceptions

Type	Condition
System.ArgumentOutOfRangeException	
System.ArgumentNullException	
System.ArgumentException	

Implements

[ICoordinate](#)  
System.IEquatable<T>

# Namespace GTiff2Tiles.Core.Enums

## Enums

### [CoordinateSystem](#)

Supported EPSG coordinate systems

### [Interpolation](#)

Represents the interpolation algorithms

### [TileExtension](#)

Extensions of ready tiles

# Enum CoordinateSystem

Supported EPSG coordinate systems

Namespace: [GTiff2Tiles.Core.Enums](#)

Assembly: GTiff2Tiles.Core.dll

## Syntax

```
public enum CoordinateSystem
```

## Fields

Name	Description
Epsg102100	Replaced by <a href="#">Epsg3857</a>
Epsg102113	Replaced by <a href="#">Epsg3857</a>
Epsg3587	Replaced by <a href="#">Epsg3857</a>
Epsg3785	Replaced by <a href="#">Epsg3857</a>
Epsg3857	EPSG:3857
Epsg41001	Replaced by <a href="#">Epsg3857</a>
Epsg4326	EPSG:4326
Epsg54004	Replaced by <a href="#">Epsg3857</a>
Epsg900913	Replaced by <a href="#">Epsg3857</a>
Other	Other or unknown coordinate system

# Enum Interpolation

Represents the interpolation algorithms

Namespace: [GTiff2Tiles.Core.Enums](#)

Assembly: GTiff2Tiles.Core.dll

## Syntax

```
public enum Interpolation
```

## Fields

Name	Description
Cubic	Cubic interpolation
Lanczos2	Two-lobe Lanczos
Lanczos3	Three-lobe Lanczos
Linear	Linear interpolation
Mitchell	Mitchell
Nearest	Nearest-neighbour interpolation

# Enum TileExtension

Extensions of ready tiles

Namespace: [GTiff2Tiles.Core.Enums](#)

Assembly: GTiff2Tiles.Core.dll

## Syntax

```
public enum TileExtension
```

## Fields

Name	Description
------	-------------

Jpg	.jpg
-----	------

Png	.png
-----	------

Webp	.webp
------	-------

# Namespace GTiff2Tiles.Core.Exceptions

## Classes

[DirectoryException](#)

[FileException](#)

[RasterException](#)



# Class DirectoryException

## Inheritance

System.Object  
System.Exception  
DirectoryException

## Implements

System.Runtime.Serialization.ISerializable

## Inherited Members

System.Exception.GetBaseException()  
System.Exception.GetObjectData(System.Runtime.Serialization.SerializationInfo, System.Runtime.Serialization.StreamingContext)  
System.Exception.GetType()  
System.Exception.ToString()  
System.Exception.Data  
System.Exception.HelpLink  
System.Exception.HResult  
System.Exception.InnerException  
System.Exception.Message  
System.Exception.Source  
System.Exception.StackTrace  
System.Exception.TargetSite  
System.Exception.SerializeObjectState  
System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [G.Tiff2Tiles.Core.Exceptions](#)

Assembly: G.Tiff2Tiles.Core.dll

## Syntax

```
public sealed class DirectoryException : Exception, ISerializable
```

## Constructors

### DirectoryException()

#### Declaration

```
public DirectoryException()
```

### DirectoryException(String)

#### Declaration

```
public DirectoryException(string message)
```

#### Parameters

Type	Name	Description
System.String	message	

### DirectoryException(String, Exception)

#### Declaration

```
public DirectoryException(string message, Exception innerException)
```

Parameters

Type	Name	Description
System.String	message	
System.Exception	innerException	

Implements

System.Runtime.Serialization.ISerializable

# Class FileNotFoundException

## Inheritance

System.Object  
System.Exception  
FileNotFoundException

## Implements

System.Runtime.Serialization.ISerializable

## Inherited Members

System.Exception.GetBaseException()  
System.Exception.GetObjectData(System.Runtime.Serialization.SerializationInfo, System.Runtime.Serialization.StreamingContext)  
System.Exception.GetType()  
System.Exception.ToString()  
System.Exception.Data  
System.Exception.HelpLink  
System.Exception.HResult  
System.Exception.InnerException  
System.Exception.Message  
System.Exception.Source  
System.Exception.StackTrace  
System.Exception.TargetSite  
System.Exception.SerializeObjectState  
System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [G.Tiff2Tiles.Core.Exceptions](#)

Assembly: G.Tiff2Tiles.Core.dll

## Syntax

```
public sealed class FileNotFoundException : Exception, ISerializable
```

## Constructors

### FileNotFoundException()

#### Declaration

```
public FileNotFoundException()
```

### FileNotFoundException(String)

#### Declaration

```
public FileNotFoundException(string message)
```

#### Parameters

Type	Name	Description
System.String	message	

### FileNotFoundException(String, Exception)

#### Declaration

```
public FileNotFoundException(string message, Exception innerException)
```

Parameters

Type	Name	Description
System.String	message	
System.Exception	innerException	

Implements

System.Runtime.Serialization.ISerializable

# Class RasterException

## Inheritance

System.Object  
System.Exception  
RasterException

## Implements

System.Runtime.Serialization.ISerializable

## Inherited Members

System.Exception.GetBaseException()  
System.Exception.GetObjectData(System.Runtime.Serialization.SerializationInfo, System.Runtime.Serialization.StreamingContext)  
System.Exception.GetType()  
System.Exception.ToString()  
System.Exception.Data  
System.Exception.HelpLink  
System.Exception.HResult  
System.Exception.InnerException  
System.Exception.Message  
System.Exception.Source  
System.Exception.StackTrace  
System.Exception.TargetSite  
System.Exception.SerializeObjectState  
System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [GTiff2Tiles.Core.Exceptions](#)

Assembly: GTiff2Tiles.Core.dll

## Syntax

```
public sealed class RasterException : Exception, ISerializable
```

## Constructors

### RasterException()

#### Declaration

```
public RasterException()
```

### RasterException(String)

#### Declaration

```
public RasterException(string message)
```

#### Parameters

Type	Name	Description
System.String	message	

### RasterException(String, Exception)

#### Declaration

```
public RasterException(string message, Exception innerException)
```

Parameters

Type	Name	Description
System.String	message	
System.Exception	innerException	

Implements

System.Runtime.Serialization.ISerializable

## Namespace GTiff2Tiles.Core.GeoTiffs

### Classes

#### [Raster](#)

Class, representing [Raster](#) GeoTiff. Used for creating [RasterTiles](#)

### Interfaces

#### [IGeoTiff](#)

Main interface for different type of GeoTiffs and [ITile](#)

# Interface IGeoTiff

Main interface for different type of GeoTiffs and [ITile](#)

## Inherited Members

System.IAsyncDisposable.DisposeAsync()  
System.IDisposable.Dispose()

Namespace: [GTiff2Tiles.Core.GeoTiffs](#)

Assembly: GTiff2Tiles.Core.dll

## Syntax

```
public interface IGeoTiff : IAsyncDisposable, IDisposable
```

## Properties

### GeoCoordinateSystem

Type of desired [CoordinateSystem](#)

#### Declaration

```
CoordinateSystem GeoCoordinateSystem { get; }
```

#### Property Value

Type	Description
<a href="#">CoordinateSystem</a>	

### IsDisposed

Shows if resources have already been disposed

#### Declaration

```
bool IsDisposed { get; }
```

#### Property Value

Type	Description
System.Boolean	

### MaxCoordinate

Maximal [GeoCoordinate](#) of this [IGeoTiff](#)

#### Declaration

```
GeoCoordinate MaxCoordinate { get; }
```

#### Property Value

Type	Description
<a href="#">GeoCoordinate</a>	

### MinCoordinate

Minimal [GeoCoordinate](#) of this [IGeoTiff](#)

#### Declaration



```
GeoCoordinate MinCoordinate { get; }
```

Property Value

Type	Description
<a href="#">GeoCoordinate</a>	

Size

Image's [Size](#) (width and height)

Declaration

```
Size Size { get; }
```

Property Value

Type	Description
<a href="#">Size</a>	

Class Raster

Class, representing [Raster](#) GeoTiff. Used for creating [RasterTiles](#)

Inheritance

System.Object  
Raster

Implements

[IGeoTiff](#)  
System.IAsyncDisposable  
System.IDisposable

Inherited Members

System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [GTile2Tiles.Core.GeoTiffs](#)

Assembly: [GTile2Tiles.Core.dll](#)

Syntax

```
public class Raster : IGeoTiff, IAsyncDisposable, IDisposable
```

Constructors

Raster(Stream, CoordinateSystem)

Declaration

```
public Raster(Stream inputStream, CoordinateSystem coordinateSystem)
```

Parameters

Type	Name	Description
System.IO.Stream	inputStream	System.IO.Stream with GeoTiff
<a href="#">CoordinateSystem</a>	coordinateSystem	

Raster(String, CoordinateSystem, Int64)

Creates new [Raster](#) object

Declaration

```
public Raster(string inputFilePath, CoordinateSystem coordinateSystem, long maxMemoryCache = 2147483648L)
```

Parameters

Type	Name	Description
System.String	inputFilePath	Input GeoTiff's path
		Must have .tif extension
<a href="#">CoordinateSystem</a>	coordinateSystem	GeoTiff's coordinate system
		If set to <a href="#">Other</a> throws System.ArgumentOutOfRangeException
System.Int64	maxMemoryCache	Max size of input image to store in RAM Must be > 0. 2GB by default

Exceptions

Type	Condition
System.ArgumentOutOfRangeException	
System.NotSupportedException	

Raster(String, Int64)

Creates new [Raster](#) object

Use this version ONLY if you don't know the [CoordinateSystem](#) of this [Raster](#). In other cases, prefer using other constructors!

Declaration

```
public Raster(string inputFilePath, long maxMemoryCache = 2147483648L)
```

Parameters

Type	Name	Description
System.String	inputFilePath	
System.Int64	maxMemoryCache	

Properties

Data

This [Raster](#)'s data

Declaration

```
public Image Data { get; }
```

Property Value

Type	Description
NetVips.Image	

GeoCoordinateSystem

Type of desired [CoordinateSystem](#)

Declaration

```
public CoordinateSystem GeoCoordinateSystem { get; }
```

Property Value

Type	Description
<a href="#">CoordinateSystem</a>	

IsDisposed

Shows if resources have already been disposed

Declaration

```
public bool IsDisposed { get; }
```

Property Value

Type	Description
System.Boolean	

MaxCoordinate

Maximal [GeoCoordinate](#) of this [IGeoTiff](#)

Declaration

```
public GeoCoordinate MaxCoordinate { get; }
```

Property Value

Type	Description
<a href="#">GeoCoordinate</a>	

MinCoordinate

Minimal [GeoCoordinate](#) of this [IGeoTiff](#)

Declaration

```
public GeoCoordinate MinCoordinate { get; }
```

Property Value

Type	Description
<a href="#">GeoCoordinate</a>	

Size

Image's [Size](#) (width and height)

Declaration

```
public Size Size { get; }
```

Property Value

Type	Description
<a href="#">Size</a>	

Methods

CreateOverviewTile(ref RasterTile, RasterTile, RasterTile, RasterTile, RasterTile, Boolean)

Creates specified overview [RasterTile](#) from 4 lower [RasterTiles](#)

Declaration

```
public static void CreateOverviewTile(ref RasterTile targetTile, RasterTile tile0, RasterTile tile1, RasterTile tile2, RasterTile tile3, bool isBuffered)
```

Parameters

Type	Name	Description
<a href="#">RasterTile</a>	targetTile	Target <a href="#">RasterTile</a>
<a href="#">RasterTile</a>	tile0	
<a href="#">RasterTile</a>	tile1	
<a href="#">RasterTile</a>	tile2	
<a href="#">RasterTile</a>	tile3	
System.Boolean	isBuffered	

Exceptions

Type	Condition
System.ArgumentNullException	

CreateOverviewTile(ref RasterTile, HashSet<RasterTile>, Boolean)

Creates specified overview [RasterTile](#) from array of lower [RasterTiles](#)

Declaration

```
public static void CreateOverviewTile(ref RasterTile targetTile, HashSet<RasterTile> baseTiles, bool isBuffered)
```

Parameters

Type	Name	Description
<a href="#">RasterTile</a>	targetTile	Target <a href="#">RasterTile</a>
System.Collections.Generic.HashSet< <a href="#">RasterTile</a> >	baseTiles	Collection of lower <a href="#">RasterTiles</a>
System.Boolean	isBuffered	Is input <a href="#">RasterTiles</a> contains data inside <a href="#">Bytes</a> property? If set to false, will use <a href="#">Path</a> to get input tiles's data

Exceptions

Type	Condition
System.ArgumentNullException	

CreateOverviewTiles(ChannelWriter<RasterTile>, Int32, Int32, HashSet<RasterTile>, Boolean, CoordinateSystem, Size, TileExtension, Boolean, Int32)

Create overview [RasterTiles](#) for specified [GeoCoordinates](#) using finding lower tiles inside System.Collections.Generic.HashSet<T> of [RasterTiles](#)

Declaration

```
public void CreateOverviewTiles(ChannelWriter<RasterTile> channelWriter, int minZ, int maxZ, HashSet<RasterTile> tiles, bool isBuffered, CoordinateSystem coordinateSystem, Size t:
```

Parameters

Type	Name	Description
System.Threading.Channels.ChannelWriter< <a href="#">RasterTile</a> >	channelWriter	System.Threading.Channels.Channel<T> to write <a href="#">RasterTiles</a>
System.Int32	minZ	Minimal overview zoom
System.Int32	maxZ	Maximal overview zoom
System.Collections.Generic.HashSet< <a href="#">RasterTile</a> >	tiles	Input <a href="#">RasterTiles</a> from which overview will be created
System.Boolean	isBuffered	Is input <a href="#">RasterTiles</a> contains data inside <a href="#">Bytes</a> property? If set to false, will use <a href="#">Path</a> to get input tiles's data
<a href="#">CoordinateSystem</a>	coordinateSystem	Target <a href="#">RasterTiles</a> coordinate system
<a href="#">Size</a>	tileSize	
<a href="#">TileExtension</a>	extension	
System.Boolean	tmsCompatible	Are ready <a href="#">RasterTiles</a> tms-compatible? false by default
System.Int32	bandsCount	

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentOutOfRangeException	

CreateOverviewTilesAsync(ChannelWriter<RasterTile>, Int32, Int32, HashSet<RasterTile>, Boolean, CoordinateSystem, Size, TileExtension, Boolean, Int32)

Create overview [RasterTiles](#) for specified [GeoCoordinates](#) using finding lower tiles inside System.Collections.Generic.HashSet<T> of [RasterTiles](#)

Declaration

```
public Task CreateOverviewTilesAsync(ChannelWriter<RasterTile> channelWriter, int minZ, int maxZ, HashSet<RasterTile> tiles, bool isBuffered, CoordinateSystem coordinateSystem, S:
```

Parameters

Type	Name	Description
System.Threading.Channels.ChannelWriter< <a href="#">RasterTile</a> >	channelWriter	System.Threading.Channels.Channel<T> to write <a href="#">RasterTiles</a>
System.Int32	minZ	Minimal overview zoom
System.Int32	maxZ	Maximal overview zoom
System.Collections.Generic.HashSet< <a href="#">RasterTile</a> >	tiles	Input <a href="#">RasterTiles</a> from which overview will be created
System.Boolean	isBuffered	Is input <a href="#">RasterTiles</a> contains data inside <a href="#">Bytes</a> property? If set to false, will use <a href="#">Path</a> to get input tiles's data
<a href="#">CoordinateSystem</a>	coordinateSystem	Target <a href="#">RasterTiles</a> coordinate system

Type	Name	Description
<a href="#">Size</a> <a href="#">TileExtension</a>	tileSize extension	
System.Boolean	tnsCompatible	Are ready <a href="#">RasterTiles</a> tns-compatible? false by default
System.Int32	bandsCount	

Returns

Type	Description
System.Threading.Tasks.Task	

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentOutOfRangeException	

CreateTileImage(Image, RasterTile)

Create NetVips.Image for one [RasterTile](#) from input NetVips.Image or tile cache

Declaration

```
public Image CreateTileImage(Image tileCache, RasterTile tile)
```

Parameters

Type	Name	Description
NetVips.Image	tileCache	Source NetVips.Image or tile cache

<a href="#">RasterTile</a>	tile	Target <a href="#">RasterTile</a>
----------------------------	------	-----------------------------------

Returns

Type	Description
NetVips.Image	Ready NetVips.Image for <a href="#">RasterTile</a>

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentException	

Dispose()

Declaration

```
public void Dispose()
```

Dispose(Boolean)

Declaration

```
protected virtual void Dispose(bool disposing)
```

Parameters

Type	Name	Description
System.Boolean	disposing	Dispose static fields?

DisposeAsync()

Declaration

```
public ValueTask DisposeAsync()
```

Returns

Type	Description
System.Threading.Tasks.ValueTask	

Finalize()

Calls [Dispose\(Boolean\)](#) on this [Raster](#)

Declaration

```
protected void Finalize()
```

GetBorders(Stream, CoordinateSystem)

Gets minimal and maximal coordinates from input GeoTiff

Declaration

```
public static (GeoCoordinate minCoordinate, GeoCoordinate maxCoordinate) GetBorders(Stream inputStream, CoordinateSystem coordinateSystem)
```

Parameters

Type	Name	Description
------	------	-------------

Type	Name	Description
System.IO.Stream	inputStream	Any kind of stream with GeoTiff's data
<a href="#">CoordinateSystem</a>	coordinateSystem	GeoTiff's <a href="#">CoordinateSystem</a> If set to <a href="#">Other</a> throws System.NotSupportedException

Returns

Type	Description
System.ValueTuple< <a href="#">GeoCoordinate</a> , <a href="#">GeoCoordinate</a> >	System.ValueTuple`2 of <a href="#">GeoCoordinates</a> of image's borders

Exceptions

Type	Condition
System.ArgumentNullException	
System.NotSupportedException	
System.ArgumentException	

GetBorders(String, CoordinateSystem)

Gets minimal and maximal coordinates from input GeoTiff

Declaration

```
public static (GeoCoordinate minCoordinate, GeoCoordinate maxCoordinate) GetBorders(string filePath, CoordinateSystem coordinateSystem)
```

Parameters

Type	Name	Description
System.String	filePath	Full path to GeoTiff file
<a href="#">CoordinateSystem</a>	coordinateSystem	

Returns

Type	Description
System.ValueTuple< <a href="#">GeoCoordinate</a> , <a href="#">GeoCoordinate</a> >	System.ValueTuple`2 of <a href="#">GeoCoordinates</a> of image's borders

Exceptions

Type	Condition
System.ArgumentNullException	
System.NotSupportedException	
System.ArgumentException	

JoinTilesIntoBytes(RasterTile, RasterTile, RasterTile, RasterTile, Boolean, Size, Int32, TileExtension)

Join 4 [RasterTiles](#) into collection of System.Bytes

if all [RasterTiles](#) are null -- returns null

Declaration

```
public static IEnumerable<byte> JoinTilesIntoBytes(RasterTile tile0, RasterTile tile1, RasterTile tile2, RasterTile tile3, bool isBuffered, Size tileSize, int bandsCount, TileExtension tileExtension)
```

Parameters

Type	Name	Description
<a href="#">RasterTile</a>	tile0	
<a href="#">RasterTile</a>	tile1	
<a href="#">RasterTile</a>	tile2	
<a href="#">RasterTile</a>	tile3	
System.Boolean	isBuffered	
<a href="#">Size</a>	tileSize	
System.Int32	bandsCount	
<a href="#">TileExtension</a>	extension	<a href="#">TileExtension</a> of ready <a href="#">RasterTile</a>

Returns

Type	Description
System.Collections.Generic.IEnumerable<System.Byte>	Collection of ready image's System.Bytes

JoinTilesIntoImage(RasterTile, RasterTile, RasterTile, RasterTile, Boolean, Size, Int32)

Join 4 [RasterTiles](#) into one NetVips.Image;

if all [RasterTiles](#) are null -- returns null

Declaration

```
public static Image JoinTilesIntoImage(RasterTile tile0, RasterTile tile1, RasterTile tile2, RasterTile tile3, bool isBuffered, Size tileSize, int bandsCount)
```

Parameters

Type	Name	Description
------	------	-------------

Type	Name	Description
<a href="#">RasterTile</a>	tile0	Upper-left <a href="#">RasterTile</a> if set to null, empty tile will be created
<a href="#">RasterTile</a>	tile1	Upper-right <a href="#">RasterTile</a> if set to null, empty tile will be created
<a href="#">RasterTile</a>	tile2	Lower-left <a href="#">RasterTile</a> if set to null, empty tile will be created
<a href="#">RasterTile</a>	tile3	Lower-right <a href="#">RasterTile</a> if set to null, empty tile will be created
System.Boolean	isBuffered	Is input <a href="#">RasterTiles</a> contains data inside <a href="#">Bytes</a> property? If set to false, will use <a href="#">Path</a> to get input tiles's data
<a href="#">Size</a>	tileSize	<a href="#">Size</a> of input and target <a href="#">RasterTile</a>
System.Int32	bandsCount	Count of bands in target <a href="#">RasterTile</a> must be in range (0-5)

Returns

Type	Description
NetVips.Image	Ready NetVips.Image

JoinTilesIntoImage(IEnumerable<Byte>, IEnumerable<Byte>, IEnumerable<Byte>, IEnumerable<Byte>, Size, Int32)

Join arrays of System.Byte of 4 [RasterTiles](#) into one NetVips.Image

if all arrays are null or empty -- returns null

Declaration

```
public static Image JoinTilesIntoImage(IEnumerable<byte> tile0Bytes, IEnumerable<byte> tile1Bytes, IEnumerable<byte> tile2Bytes, IEnumerable<byte> tile3Bytes, Size tileSize, int bandsCount)
```

Parameters

Type	Name	Description
System.Collections.Generic.IEnumerable<System.Byte>	tile0Bytes	Bytes of upper-left <a href="#">RasterTile</a> if set to null, empty tile will be created
System.Collections.Generic.IEnumerable<System.Byte>	tile1Bytes	Bytes of upper-right <a href="#">RasterTile</a> if set to null, empty tile will be created
System.Collections.Generic.IEnumerable<System.Byte>	tile2Bytes	Bytes of lower-left <a href="#">RasterTile</a> if set to null, empty tile will be created
System.Collections.Generic.IEnumerable<System.Byte>	tile3Bytes	Bytes of lower-right <a href="#">RasterTile</a> if set to null, empty tile will be created
<a href="#">Size</a>	tileSize	<a href="#">Size</a> of input and target <a href="#">RasterTile</a>
System.Int32	bandsCount	Count of bands in target <a href="#">RasterTile</a> must be in range (0-5)

Returns

Type	Description
NetVips.Image	Ready NetVips.Image

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentOutOfRangeException	

JoinTilesIntoImage(String, String, String, String, Size, Int32)

Join 4 [RasterTiles](#) into one NetVips.Image

if all 4 paths are null or empty System.Strings -- returns null

Declaration

```
public static Image JoinTilesIntoImage(string tile0Path, string tile1Path, string tile2Path, string tile3Path, Size tileSize, int bandsCount)
```

Parameters

Type	Name	Description
------	------	-------------

Type	Name	Description
System.String	tile0Path	Path of upper-left <a href="#">RasterTile</a> if set to null, empty tile will be created
	tile1Path	Path of upper-right <a href="#">RasterTile</a> if set to null, empty tile will be created
System.String	tile2Path	Path of lower-left <a href="#">RasterTile</a> if set to null, empty tile will be created
	tile3Path	Path of lower-right <a href="#">RasterTile</a> if set to null, empty tile will be created
<a href="#">Size</a>	tileSize	<a href="#">Size</a> of input and target <a href="#">RasterTile</a>
System.Int32	bandsCount	Count of bands in target <a href="#">RasterTile</a> must be in range (0-5)

Returns

Type	Description
NetVips.Image	Ready NetVips.Image

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentOutOfRangeException	

JoinTilesIntoImageAsync(RasterTile, RasterTile, RasterTile, RasterTile, Boolean, Size, Int32)

Join 4 [RasterTiles](#) into one NetVips.Image;  
if all [RasterTiles](#) are null -- returns null

Declaration

```
public static Task<Image> JoinTilesIntoImageAsync(RasterTile tile0, RasterTile tile1, RasterTile tile2, RasterTile tile3, bool isBuffered, Size tileSize, int bandsCount)
```

Parameters

Type	Name	Description
<a href="#">RasterTile</a>	tile0	Upper-left <a href="#">RasterTile</a> if set to null, empty tile will be created
	tile1	Upper-right <a href="#">RasterTile</a> if set to null, empty tile will be created
<a href="#">RasterTile</a>	tile2	Lower-left <a href="#">RasterTile</a> if set to null, empty tile will be created
	tile3	Lower-right <a href="#">RasterTile</a> if set to null, empty tile will be created
System.Boolean	isBuffered	Is input <a href="#">RasterTiles</a> contains data inside <a href="#">Bytes</a> property? If set to false, will use <a href="#">Path</a> to get input tiles's data
	tileSize	<a href="#">Size</a> of input and target <a href="#">RasterTile</a>
System.Int32	bandsCount	Count of bands in target <a href="#">RasterTile</a> must be in range (0-5)

Returns

Type	Description
System.Threading.Tasks.Task<NetVips.Image>	Ready NetVips.Image

JoinTilesIntoImageAsync(IEnumerable<Byte>, IEnumerable<Byte>, IEnumerable<Byte>, IEnumerable<Byte>, Size, Int32)

Join arrays of System.Byte of 4 [RasterTiles](#) into one NetVips.Image  
if all arrays are null or empty -- returns null

Declaration

```
public static Task<Image> JoinTilesIntoImageAsync(IEnumerable<byte> tile0Bytes, IEnumerable<byte> tile1Bytes, IEnumerable<byte> tile2Bytes, IEnumerable<byte> tile3Bytes, Size tileSize, int bandsCount)
```

Parameters

Type	Name	Description
------	------	-------------



Type	Name	Description
System.Collections.Generic.IEnumerable<System.Byte> tile0Bytes		Bytes of upper-left <a href="#">RasterTile</a>
		if set to null, empty tile will be created
System.Collections.Generic.IEnumerable<System.Byte> tile1Bytes		Bytes of upper-right <a href="#">RasterTile</a>
		if set to null, empty tile will be created
System.Collections.Generic.IEnumerable<System.Byte> tile2Bytes		Bytes of lower-left <a href="#">RasterTile</a>
		if set to null, empty tile will be created
System.Collections.Generic.IEnumerable<System.Byte> tile3Bytes		Bytes of lower-right <a href="#">RasterTile</a>
		if set to null, empty tile will be created
<a href="#">Size</a>	tileSize	<a href="#">Size</a> of input and target <a href="#">RasterTile</a>
System.Int32	bandsCount	Count of bands in target <a href="#">RasterTile</a> must be in range (0-5)

Returns

Type	Description
System.Threading.Tasks.Task<NetVips.Image>	Ready NetVips.Image

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentOutOfRangeException	

JoinTilesIntoImageAsync(String, String, String, String, Size, Int32)

Join 4 [RasterTiles](#) into one NetVips.Image

if all 4 paths are null or empty System.Strings -- returns null

Declaration

```
public static Task<Image> JoinTilesIntoImageAsync(string tile0Path, string tile1Path, string tile2Path, string tile3Path, Size tileSize, int bandsCount)
```

Parameters

Type	Name	Description
System.String tile0Path		Path of upper-left <a href="#">RasterTile</a>
		if set to null, empty tile will be created
System.String tile1Path		Path of upper-right <a href="#">RasterTile</a>
		if set to null, empty tile will be created
System.String tile2Path		Path of lower-left <a href="#">RasterTile</a>
		if set to null, empty tile will be created
System.String tile3Path		Path of lower-right <a href="#">RasterTile</a>
		if set to null, empty tile will be created
<a href="#">Size</a>	tileSize	<a href="#">Size</a> of input and target <a href="#">RasterTile</a>
System.Int32	bandsCount	Count of bands in target <a href="#">RasterTile</a> must be in range (0-5)

Returns

Type	Description
System.Threading.Tasks.Task<NetVips.Image>	Ready NetVips.Image

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentOutOfRangeException	

WriteTilesToAsyncEnumerable(Int32, Int32, Boolean, Size, Interpolation, Int32, Int32, Int32, IProgress<Double>, Action<String>)

Crops current [Raster](#) on [RasterTiles](#) and writes them to System.Collections.Generic.IAsyncEnumerable<T>

Declaration

```
public IAsyncEnumerable<RasterTile> WriteTilesToAsyncEnumerable(int minZ, int maxZ, bool tmsCompatible = false, Size tileSize = null, Interpolation interpolation = Interpolation.)
```

Parameters

Type	Name	Description
------	------	-------------

Type	Name	Description
System.Int32	minZ	Minimum cropped zoom Should be >= 0 and lesser or equal, than <code>maxZ</code>
System.Int32	maxZ	Maximum cropped zoom Should be >= 0 and bigger or equal, than <code>minZ</code>
System.Boolean	tmsCompatible	Do you want to create tms-compatible <a href="#">ITiles</a> ? false by default
<a href="#">Size</a>	tileSize	<a href="#">Size</a> of <a href="#">ITiles</a> 256x256 by default
<a href="#">Interpolation</a>	interpolation	Interpolation of ready tiles <a href="#">Lanczos3</a> by default
System.Int32	bandsCount	Count of <a href="#">Bands</a> in ready <a href="#">ITiles</a> 4 by default
System.Int32	tileCacheCount	Count of <a href="#">ITile</a> to be in cache 1000 by default
System.Int32	threadsCount	Threads count Calculates automatically by default
System.IProgress<System.Double>	progress	Progress-reporter null by default
System.Action<System.String>	printTimeAction	System.Action<T> to print estimated time null by default; set to null if you don't want output

Returns

Type	Description
System.Collections.Generic.IAsyncEnumerable< <a href="#">RasterTile</a> >	System.Collections.Generic.IAsyncEnumerable<T> of <a href="#">RasterTiles</a>

Exceptions

Type	Condition
System.ArgumentOutOfRangeException	
<a href="#">RasterException</a>	

**WriteTilesToChannel(ChannelWriter<[RasterTile](#)>, Int32, Int32, Boolean, Size, Interpolation, Int32, Int32, Int32, IProgress<Double>, Action<String>)**

Crops current [Raster](#) on [RasterTiles](#) and writes them to `channelWriter`

Declaration

```
public void WriteTilesToChannel(ChannelWriter<RasterTile> channelWriter, int minZ, int maxZ, bool tmsCompatible = false, Size tileSize = null, Interpolation interpolation = Interpolation.Lanczos3, int bandsCount = 4, int tileCacheCount = 1000, int threadsCount = 1, IProgress<double> progress = null, Action<string> printTimeAction = null)
```

Parameters

Type	Name	Description
System.Threading.Channels.ChannelWriter< <a href="#">RasterTile</a> >	channelWriter	System.Threading.Channels.Channel to write <a href="#">RasterTile</a> to
System.Int32	minZ	
System.Int32	maxZ	
System.Boolean	tmsCompatible	
<a href="#">Size</a>	tileSize	
<a href="#">Interpolation</a>	interpolation	
System.Int32	bandsCount	
System.Int32	tileCacheCount	
System.Int32	threadsCount	
System.IProgress<System.Double>	progress	
System.Action<System.String>	printTimeAction	

Exceptions

Type	Condition
System.ArgumentOutOfRangeException	
<a href="#">RasterException</a>	

**WriteTilesToChannelAsync(ChannelWriter<[RasterTile](#)>, Int32, Int32, Boolean, Size, Interpolation, Int32, Int32, Int32, IProgress<Double>, Action<String>)**

Crops current [Raster](#) on [RasterTiles](#) and writes them to `channelWriter`

Declaration

```
public Task WriteTilesToChannelAsync(ChannelWriter<RasterTile> channelWriter, int minZ, int maxZ, bool tmsCompatible = false, Size tileSize = null, Interpolation interpolation = Interpolation.Lanczos3, int bandsCount = 4, int tileCacheCount = 1000, int threadsCount = 1, IProgress<double> progress = null, Action<string> printTimeAction = null)
```

Parameters

Type	Name	Description
System.Threading.Channels.ChannelWriter< <a href="#">RasterTile</a> >	channelWriter	System.Threading.Channels.Channel to write <a href="#">RasterTile</a> to
System.Int32	minZ	
System.Int32	maxZ	
System.Boolean	tmsCompatible	
<a href="#">Size</a>	tileSize	
<a href="#">Interpolation</a>	interpolation	
System.Int32	bandsCount	
System.Int32	tileCacheCount	
System.Int32	threadsCount	
System.IProgress<System.Double>	progress	
System.Action<System.String>	printTimeAction	

Returns

Type	Description
System.Threading.Tasks.Task	

Exceptions

Type	Condition
System.ArgumentOutOfRangeException	
<a href="#">RasterException</a>	

**WriteTilesToDirectory(String, Int32, Int32, Boolean, Size, TileExtension, Interpolation, Int32, Int32, Int32, IProgress<Double>, Action<String>)**

Crops current [RasterTile](#) on [RasterTiles](#) and writes them to outputDirectoryPath

Declaration

```
public void WriteTilesToDirectory(string outputDirectoryPath, int minZ, int maxZ, bool tmsCompatible = false, Size tileSize = null, TileExtension tileExtension = TileExtension.Png
```

Parameters

Type	Name	Description
System.String	outputDirectoryPath	Directory for output <a href="#">RasterTiles</a>
System.Int32	minZ	
System.Int32	maxZ	
System.Boolean	tmsCompatible	
<a href="#">Size</a>	tileSize	
<a href="#">TileExtension</a>	tileExtension	Extension of ready <a href="#">RasterTiles</a> .png by default
<a href="#">Interpolation</a>	interpolation	
System.Int32	bandsCount	
System.Int32	tileCacheCount	
System.Int32	threadsCount	T
System.IProgress<System.Double>	progress	
System.Action<System.String>	printTimeAction	

Exceptions

Type	Condition
System.ArgumentOutOfRangeException	
<a href="#">RasterException</a>	

**WriteTilesToDirectoryAsync(String, Int32, Int32, Boolean, Size, TileExtension, Interpolation, Int32, Int32, Int32, IProgress<Double>, Action<String>)**

Crops current [RasterTile](#) on [RasterTiles](#) and writes them to outputDirectoryPath

Declaration

```
public Task WriteTilesToDirectoryAsync(string outputDirectoryPath, int minZ, int maxZ, bool tmsCompatible = false, Size tileSize = null, TileExtension tileExtension = TileExtensio
```

Parameters

Type	Name	Description
System.String	outputDirectoryPath	Directory for output <a href="#">RasterTiles</a>
System.Int32	minZ	
System.Int32	maxZ	
System.Boolean	tmsCompatible	
<a href="#">Size</a>	tileSize	
<a href="#">TileExtension</a>	tileExtension	Extension of ready <a href="#">RasterTiles</a> .png by default
<a href="#">Interpolation</a>	interpolation	
System.Int32	bandsCount	
System.Int32	tileCacheCount	
System.Int32	threadsCount	T
System.IProgress<System.Double>	progress	
System.Action<System.String>	printTimeAction	

Returns

Type	Description
System.Threading.Tasks.Task	

Exceptions

Type	Condition
System.ArgumentOutOfRangeException	
<a href="#">RasterException</a>	

**WriteTilesToEnumerable(Int32, Int32, Boolean, Size, Interpolation, Int32, Int32, IProgress<Double>, Action<String>)**

Crops current [Raster](#) on [RasterTiles](#) and writes them to System.Collections.Generic.IEnumerable<T>

Declaration

```
public IEnumerable<RasterTile> WriteTilesToEnumerable(int minZ, int maxZ, bool tmsCompatible = false, Size tileSize = null, Interpolation interpolation = Interpolation.Lanczos3, :
```

Parameters

Type	Name	Description
System.Int32	minZ	
System.Int32	maxZ	
System.Boolean	tmsCompatible	
<a href="#">Size</a>	tileSize	
<a href="#">Interpolation</a>	interpolation	
System.Int32	bandsCount	
System.Int32	tileCacheCount	
System.IProgress<System.Double>	progress	
System.Action<System.String>	printTimeAction	

Returns

Type	Description
System.Collections.Generic.IEnumerable< <a href="#">RasterTile</a> >	System.Collections.Generic.IEnumerable<T> of <a href="#">RasterTiles</a>

Exceptions

Type	Condition
System.ArgumentOutOfRangeException	
<a href="#">RasterException</a>	

**WriteTileToChannel(Image, RasterTile, ChannelWriter<RasterTile>)**

Gets data from source NetVips.Image or tile cache for specified [RasterTile](#) and writes it to System.Threading.Channels.ChannelWriter<T>

Declaration

```
public bool WriteTileToChannel(Image tileCache, RasterTile tile, ChannelWriter<RasterTile> channelWriter)
```

Parameters

Type	Name	Description
NetVips.Image	tileCache	Source NetVips.Image or tile cache
<a href="#">RasterTile</a>	tile	Target <a href="#">RasterTile</a>

System.Threading.Channels.ChannelWriter<[RasterTile](#)> channelWriter Target System.Threading.Channels.ChannelWriter<T>

Returns

Type	Description
System.Boolean	true if <a href="#">RasterTile</a> was written; false otherwise

Exceptions

Type	Condition
System.ArgumentNullException	

**WriteTileToChannelAsync(Image, RasterTile, ChannelWriter<RasterTile>)**

Gets data from source NetVips.Image or tile cache for specified [RasterTile](#) and writes it to System.Threading.Channels.ChannelWriter<T>

Declaration

```
public ValueTask WriteTileToChannelAsync(Image tileCache, RasterTile tile, ChannelWriter<RasterTile> channelWriter)
```

Parameters

Type	Name	Description
NetVips.Image	tileCache	Source NetVips.Image or tile cache
<a href="#">RasterTile</a>	tile	Target <a href="#">RasterTile</a>

System.Threading.Channels.ChannelWriter<[RasterTile](#)> channelWriter Target System.Threading.Channels.ChannelWriter<T>

Returns

Type	Description
System.Threading.Tasks.ValueTask	

Exceptions

Type	Condition
System.ArgumentNullException	

WriteTileToEnumerable(Image, RasterTile)

Gets data from source NetVips.Image or tile cache for specified [RasterTile](#) and writes it to System.Collections.Generic.IEnumerable<T>

Declaration

```
public IEnumerable<byte> WriteTileToEnumerable(Image tileCache, RasterTile tile)
```

Parameters

Type	Name	Description
NetVips.Image	tileCache	Source NetVips.Image or tile cache
<a href="#">RasterTile</a>	tile	Target <a href="#">RasterTile</a>

Returns

Type	Description
System.Collections.Generic.IEnumerable<System.Byte>	<a href="#">RasterTile</a> 's System.Bytes

WriteTileToFile(Image, RasterTile)

Gets data from source NetVips.Image or tile cache for specified [RasterTile](#) and writes it to ready file

Declaration

```
public void WriteTileToFile(Image tileCache, RasterTile tile)
```

Parameters

Type	Name	Description
NetVips.Image	tileCache	Source NetVips.Image or tile cache
<a href="#">RasterTile</a>	tile	Target <a href="#">RasterTile</a> <a href="#">Path</a> should not be null or whitespace

Exceptions

Type	Condition
System.ArgumentNullException	

WriteTileToFileAsync(Image, RasterTile)

Gets data from source NetVips.Image or tile cache for specified [RasterTile](#) and writes it to ready file

Declaration

```
public Task WriteTileToFileAsync(Image tileCache, RasterTile tile)
```

Parameters

Type	Name	Description
NetVips.Image	tileCache	Source NetVips.Image or tile cache
<a href="#">RasterTile</a>	tile	Target <a href="#">RasterTile</a> <a href="#">Path</a> should not be null or whitespace

Returns

Type	Description
System.Threading.Tasks.Task	

Exceptions

Type	Condition
System.ArgumentNullException	

Implements

[IGeoTiff](#)  
System.IAsyncDisposable  
System.IDisposable

# Namespace GTiff2Tiles.Core.Helpers

## Classes

### [CheckHelper](#)

Class with static methods to check for errors

### [NetVipsHelper](#)

Some additional methods for NetVips

### [ProgressHelper](#)

Class with methods to simplify progress-reporting

# Class CheckHelper

Class with static methods to check for errors

## Inheritance

System.Object  
CheckHelper

## Inherited Members

System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [G:Tiff2Tiles.Core.Helpers](#)

Assembly: G:Tiff2Tiles.Core.dll

## Syntax

```
public static class CheckHelper
```

## Methods

### CheckDirectory(String, Nullable<Boolean>)

Checks, if directory's path is not empty, creates directory if it doesn't exist and checks if it's empty or not

#### Declaration

```
public static void CheckDirectory(string directoryPath, bool? shouldBeEmpty = default(bool? ))
```

#### Parameters

Type	Name	Description
System.String	directoryPath	Directory's path to check
System.Nullable<System.Boolean>	shouldBeEmpty	Should directory be empty? If set , emptyness doesn't check

#### Exceptions

Type	Condition
System.ArgumentNullException	
<a href="#">DirectoryException</a>	

### CheckFile(String, Nullable<Boolean>, String)

Checks, if file's path is not empty string and file exists, if it should

#### Declaration

```
public static void CheckFile(string filePath, bool? shouldExist, string fileExtension = null)
```

#### Parameters

Type	Name	Description
System.String	filePath	File's path to check
		Should the file exist?
System.Nullable<System.Boolean>	shouldExist	true by default; set this to null if you don't know or care if file's already exists
System.String	fileExtension	Checks file extension If set to , extension doesn't check

#### Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentException	
System.IO.FileNotFoundException	
<a href="#">FileException</a>	

#### CheckInputFileAsync(String, CoordinateSystem)

Checks the existence, projection and type

#### Declaration

```
public static ValueTask<bool> CheckInputFileAsync(string inputFilePath, CoordinateSystem targetSystem)
```

#### Parameters

Type	Name	Description
System.String	inputFilePath	Input GeoTiff's path
<a href="#">CoordinateSystem</a>	targetSystem	Target coordinate system

#### Returns

Type	Description
System.Threading.Tasks.ValueTask<System.Boolean>	true if file needs to be converted; false otherwise



# Class NetVipsHelper

Some additional methods for NetVips

## Inheritance

System.Object  
NetVipsHelper

## Inherited Members

System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [GTiff2Tiles.Core.Helpers](#)

Assembly: GTiff2Tiles.Core.dll

## Syntax

```
public static class NetVipsHelper
```

## Methods

### DisableLog()

Disables NetVips log warnings

## Declaration

```
public static void DisableLog()
```

# Class ProgressHelper

Class with methods to simplify progress-reporting

## Inheritance

System.Object  
ProgressHelper

## Inherited Members

System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [G TIFF2 Tiles.Core.Helpers](#)

Assembly: G TIFF2 Tiles.Core.dll

## Syntax

```
public static class ProgressHelper
```

## Methods

### GetEstimatedTimeLeft(Double, Stopwatch)

Calculate estimated time left, based on your current progress and time from start

#### Declaration

```
public static TimeSpan GetEstimatedTimeLeft(double percentage, Stopwatch stopwatch)
```

#### Parameters

Type	Name	Description
System.Double	percentage	Current progress; Should be in range (0.0, 100.0]

System.Diagnostics.Stopwatch stopwatch Time passed from the start

#### Returns

Type	Description
System.TimeSpan	Estimated System.TimeSpan left

#### Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentOutOfRangeException	

### PrintEstimatedTimeLeft(Double, Stopwatch, Action<String>)

Prints estimated time left

#### Declaration

```
public static void PrintEstimatedTimeLeft(double percentage, Stopwatch stopwatch = null, Action<string> reporter = null)
```

Parameters

Type	Name	Description
System.Double	percentage	Current progress; Should be in range (0.0, 100.0]
System.Diagnostics.Stopwatch	stopwatch	Time passed from the start; If set to null no time printed
System.Action<System.String>	reporter	Delegate to work with reported string E.g. System.Console.WriteLine(System.String); if set to null no time printed

# Namespace GTiff2Tiles.Core.Images

## Classes

### [Area](#)

Represents read/write [Areas](#) of image

### [Band](#)

Represents image's band

### [Size](#)

[Size](#) of any image

Class Area

Represents read/write [Areas](#) of image

Inheritance

System.Object  
Area

Inherited Members

System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [Gtiff2Tiles.Core.Images](#)

Assembly: Gtiff2Tiles.Core.dll

Syntax

public class Area

Constructors

**Area(PixelCoordinate, Size)**

Creates new [Area](#)

Declaration

public Area(PixelCoordinate originCoordinate, Size size)

Parameters

Type	Name	Description
<a href="#">PixelCoordinate</a>	originCoordinate	<a href="#">OriginCoordinate</a>
<a href="#">Size</a>	size	<a href="#">Size</a>

Exceptions

Type	Condition
System.ArgumentNullException	

Properties

**OriginCoordinate**

Origin [PixelCoordinate](#)

Declaration

public PixelCoordinate OriginCoordinate { get; }

Property Value

Type	Description
<a href="#">PixelCoordinate</a>	

**Size**

[Size](#) of [Area](#)

Declaration

public Size Size { get; }

Property Value

Type	Description
<a href="#">Size</a>	

Methods

**GetAreas(GeoCoordinate, GeoCoordinate, Size, GeoCoordinate, GeoCoordinate, Size)**

Get [Areas](#) to read from input [IGeoTiff](#) and to write to target [ITile](#)

Declaration

public static (Area readArea, Area writeArea) GetAreas(GeoCoordinate imageMinCoordinate, GeoCoordinate imageMaxCoordinate, Size imageSize, GeoCoordinate tileMinCoordinate, GeoCoo:

Parameters

Type	Name	Description
<a href="#">GeoCoordinate</a>	imageMinCoordinate	Minimal <a href="#">GeoCoordinate</a> of <a href="#">IGeoTiff</a>
<a href="#">GeoCoordinate</a>	imageMaxCoordinate	Maximal <a href="#">GeoCoordinate</a> of <a href="#">IGeoTiff</a>
<a href="#">Size</a>	imageSize	<a href="#">Size</a> of <a href="#">IGeoTiff</a>
<a href="#">GeoCoordinate</a>	tileMinCoordinate	Minimal <a href="#">GeoCoordinate</a> of <a href="#">ITile</a>
<a href="#">GeoCoordinate</a>	tileMaxCoordinate	Maximal <a href="#">GeoCoordinate</a> of <a href="#">ITile</a>

Type	Name	Description
<a href="#">Size</a>	tileSize	<a href="#">Size</a> of <a href="#">ITile</a>

Returns

Type	Description
System.ValueTuple< <a href="#">Area</a> , <a href="#">Area</a> >	

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentException	

**GetAreas(IGeoTiff, ITile)**

Get [Areas](#) to read from input [IGeoTiff](#) and to write to target [ITile](#).

Declaration

```
public static (Area readArea, Area writeArea) GetAreas(IGeoTiff image, ITile tile)
```

Parameters

Type	Name	Description
<a href="#">IGeoTiff</a> image	Source <a href="#">IGeoTiff</a>	
<a href="#">ITile</a> tile	Target <a href="#">ITile</a>	

Returns

Type	Description
System.ValueTuple< <a href="#">Area</a> , <a href="#">Area</a> >	System.ValueTuple`2 of <a href="#">Areas</a> to read and write

Exceptions

Type	Condition
System.ArgumentNullException	

# Class Band

Represents image's band

## Inheritance

System.Object  
Band

## Inherited Members

System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [GTiff2Tiles.Core.Images](#)

Assembly: GTiff2Tiles.Core.dll

## Syntax

```
public class Band
```

## Constructors

### Band(Int32)

Creates new [Band](#)

## Declaration

```
public Band(int value = 255)
```

## Parameters

Type	Name	Description
System.Int32	value	System.Int32 in range from 0 to 255

## Exceptions

Type	Condition
System.ArgumentOutOfRangeException	

## Fields

### DefaultValue

Default [Band](#) value

## Declaration

```
public const int DefaultValue = 255
```

## Field Value

Type	Description
System.Int32	

# Properties

## Value

Current value

## Declaration

```
public int Value { get; }
```

## Property Value

Type	Description
System.Int32	

# Methods

## AddBands(ref Image, IEnumerable<Band>)

Add [Bands](#) to NetVips.Image

## Declaration

```
public static void AddBands(ref Image image, IEnumerable<Band> bands)
```

## Parameters

Type	Name	Description
NetVips.Image	image	Reference on NetVips.Image to add <a href="#">Bands</a> to
System.Collections.Generic.IEnumerable< <a href="#">Band</a> >	bands	Collection of <a href="#">Bands</a> to add

## Exceptions

Type	Condition
System.ArgumentNullException	

## AddDefaultBands(ref Image, Int32)

Add default [Bands](#) to NetVips.Image until bands count is lesser than NetVips.Image's current bands count

## Declaration

```
public static void AddDefaultBands(ref Image image, int bandsCount)
```

## Parameters

Type	Name	Description
NetVips.Image	image	Reference on NetVips.Image to add <a href="#">Bands</a> to
System.Int32	bandsCount	Count of desired <a href="#">Bands</a> in NetVips.Image; NOT the count of <a href="#">Bands</a> to add

## Exceptions

Type	Condition
------	-----------



Type	Condition
System.ArgumentNullException	

# Class Size

[Size](#) of any image

## Inheritance

System.Object  
Size

## Implements

System.IEquatable<[Size](#)>

## Inherited Members

System.Object.Equals(System.Object, System.Object)  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [G.Tiff2Tiles.Core.Images](#)

Assembly: G.Tiff2Tiles.Core.dll

## Syntax

```
public sealed class Size : IEquatable<Size>
```

## Constructors

### Size(Int32, Int32)

Creates new [Size](#)

## Declaration

```
public Size(int width, int height)
```

## Parameters

Type	Name	Description
System.Int32	<a href="#">Width</a>	width Should be > 0
System.Int32	<a href="#">Height</a>	height Should be > 0

## Exceptions

Type	Condition
System.ArgumentOutOfRangeException	

## Properties

### Height

Image's height

## Declaration

```
public int Height { get; }
```

Property Value

Type	Description
System.Int32	

IsSquare

Shows if this tile is square (width == height)

Declaration

```
public bool IsSquare { get; }
```

Property Value

Type	Description
System.Boolean	

Resolution

Image's resolution

Declaration

```
public int Resolution { get; }
```

Property Value

Type	Description
System.Int32	

Width

Image's width

Declaration

```
public int Width { get; }
```

Property Value

Type	Description
System.Int32	

Methods

Add(Size)

Sum [Sizes](#)

Declaration

```
public Size Add(Size other)
```

Parameters

Type	Name	Description
------	------	-------------

[Size](#) other [Size](#) to add

Returns

Type Description

[Size](#) New [Size](#)

Exceptions

Type	Condition
System.ArgumentNullException	

Divide(Size)

Divide [Size](#)s

Declaration

```
public Size Divide(Size other)
```

Parameters

Type Name Description

[Size](#) other [Size](#) to divide on

Returns

Type Description

[Size](#) New [Size](#)

Exceptions

Type	Condition
System.ArgumentNullException	

Equals(Size)

Declaration

```
public bool Equals(Size other)
```

Parameters

Type Name Description

[Size](#) other

Returns

Type	Description
System.Boolean	

Equals(Object)

Declaration

```
public override bool Equals(object size)
```

Parameters

Type	Name	Description
System	Object	size

Returns

Type	Description
System	Boolean

Overrides

System.Object.Equals(System.Object)

GetHashCode()

Declaration

```
public override int GetHashCode()
```

Returns

Type	Description
System	Int32

Overrides

System.Object.GetHashCode()

Multiply(Size)

Multiply [Sizes](#)

Declaration

```
public Size Multiply(Size other)
```

Parameters

Type	Name	Description
<a href="#">Size</a>	other	<a href="#">Size</a> to multiply

Returns

Type Description

[Size](#) New [Size](#)

Exceptions

Type	Condition
System	ArgumentNullException

Subtract(Size)

Subtract [Sizes](#)

Declaration

```
public Size Subtract(Size other)
```

Parameters

**Type Name    Description**

[Size](#)   other   [Size](#) to subtract

**Returns**

**Type Description**

[Size](#)   New [Size](#)

**Exceptions**

Type	Condition
System.ArgumentNullException	

**Operators**

**Addition([Size](#), [Size](#))**

Sum [Sizes](#)

**Declaration**

```
public static Size operator +(Size size1, Size size2)
```

**Parameters**

**Type Name Description**

[Size](#)   size1   [Size](#) 1

[Size](#)   size2   [Size](#) 2

**Returns**

**Type Description**

[Size](#)   New [Size](#)

**Exceptions**

Type	Condition
System.ArgumentNullException	

**Division([Size](#), [Size](#))**

Divide [Sizes](#)

**Declaration**

```
public static Size operator /(Size size1, Size size2)
```

**Parameters**

**Type Name Description**

[Size](#)   size1   [Size](#) 1

**Type Name Description**

[Size](#) size2 [Size](#) 2

**Returns**

**Type Description**

[Size](#) New [Size](#)

**Exceptions**

Type	Condition
System.ArgumentNullException	

**Equality(Size, Size)**

Check two [Sizes](#) for equality

**Declaration**

```
public static bool operator ==(Size size1, Size size2)
```

**Parameters**

**Type Name Description**

[Size](#) size1 [Size](#) 1

[Size](#) size2 [Size](#) 2

**Returns**

Type	Description
System.Boolean	true if <a href="#">Sizes</a> are equal; false otherwise

**Inequality(Size, Size)**

Check two [Sizes](#) for non-equality

**Declaration**

```
public static bool operator !=(Size size1, Size size2)
```

**Parameters**

**Type Name Description**

[Size](#) size1 [Size](#) 1

[Size](#) size2 [Size](#) 2

**Returns**

Type	Description
------	-------------

Type	Description
System.Boolean	true if <a href="#">Sizes</a> are not equal; false otherwise

Multiply(Size, Size)

Multiply [Sizes](#)

Declaration

```
public static Size operator *(Size size1, Size size2)
```

Parameters

Type Name Description

[Size](#) size1 [Size](#) 1

[Size](#) size2 [Size](#) 2

Returns

Type Description

[Size](#) New [Size](#)

Exceptions

Type	Condition
System.ArgumentNullException	

Subtraction(Size, Size)

Subtract [Sizes](#)

Declaration

```
public static Size operator -(Size size1, Size size2)
```

Parameters

Type Name Description

[Size](#) size1 [Size](#) 1

[Size](#) size2 [Size](#) 2

Returns

Type Description

[Size](#) New [Size](#)

Exceptions

Type	Condition
------	-----------



Type	Condition
System.ArgumentNullException	

## Implements

System.IEquatable<T>

# Namespace GTiff2Tiles.Core.Tiles

## Classes

### [Number](#)

[Number](#) of [ITile](#)

### [RasterTile](#)

[Raster Tile](#)

### [Tile](#)

Basic implementation of [ITile](#) interface

## Interfaces

### [ITile](#)

Interface for all tiles

# Interface ITile

Interface for all tiles

## Inherited Members

System.IDisposable.Dispose()  
System.IAsyncDisposable.DisposeAsync()

Namespace: [GTiff2Tiles.Core.Tiles](#)

Assembly: GTiff2Tiles.Core.dll

## Syntax

```
public interface ITile : IDisposable, IAsyncDisposable
```

## Properties

### Bytes

Collection of [ITile](#)'s bytes

#### Declaration

```
IEnumerable<byte> Bytes { get; set; }
```

#### Property Value

Type	Description
System.Collections.Generic.IEnumerable<System.Byte>	

## Extension

Extension of [ITile](#)

#### Declaration

```
TileExtension Extension { get; }
```

#### Property Value

Type	Description
<a href="#">TileExtension</a>	

## IsDisposed

Shows if this [ITile](#)'s already disposed

#### Declaration

```
bool IsDisposed { get; }
```

#### Property Value

Type	Description
System.Boolean	

## MaxCoordinate

Maximal [GeoCoordinate](#) of this [ITile](#)

#### Declaration

```
GeoCoordinate MaxCoordinate { get; }
```

#### Property Value

Type	Description
<a href="#">GeoCoordinate</a>	

#### MinCoordinate

Minimal [GeoCoordinate](#) of this [ITile](#)

#### Declaration

```
GeoCoordinate MinCoordinate { get; }
```

#### Property Value

Type	Description
<a href="#">GeoCoordinate</a>	

#### MinimalBytesCount

[ITiles](#) with [Bytes](#) count lesser than this value won't pass [Validate\(Boolean\)](#) check

#### Declaration

```
int MinimalBytesCount { get; set; }
```

#### Property Value

Type	Description
System.Int32	

#### Number

[Number](#) of this [ITile](#)

#### Declaration

```
Number Number { get; }
```

#### Property Value

Type	Description
<a href="#">Number</a>	

#### Path

Path on disk of this [ITile](#)

#### Declaration

```
string Path { get; set; }
```

#### Property Value

Type	Description
System.String	

#### Size

[Size](#) (width and height) of this [ITile](#)

## Declaration

```
Size Size { get; }
```

## Property Value

## Type Description

[Size](#)

## TmsCompatible

Is [ITile](#) tms compatible?

## Declaration

```
bool TmsCompatible { get; }
```

## Property Value

Type	Description
System.Boolean	

## Methods

### CalculatePosition()

Calculates this [ITile](#)'s position in upper [ITile](#)

## Declaration

```
int CalculatePosition()
```

## Returns

Type	Description
System.Int32	Value in range from 0 to 3
	Starts always from upper-left corner and goes to lower-right, but maths depends on <a href="#">TmsCompatible</a> value

### GetExtensionString()

Get System.String from [TileExtension](#)

## Declaration

```
string GetExtensionString()
```

## Returns

Type	Description
System.String	Converted System.String

### Validate(Boolean)

Checks if this [ITile](#) is not empty or too small

See [MinimalBytesCount](#) property for more info

## Declaration

```
bool Validate(bool isCheckPath)
```

#### Parameters

Type	Name	Description
System.Boolean	isCheckPath	Do you want to check <a href="#">Path</a> ?

#### Returns

Type	Description
System.Boolean	true if <a href="#">ITile</a> 's valid; false otherwise

#### WriteToFile(String)

Writes [ITile](#)'s [Bytes](#) to file

#### Declaration

```
void WriteToFile(string path = null)
```

#### Parameters

Type	Name	Description
System.String	path	Full path to write <a href="#">ITile</a> if not set, <a href="#">Path</a> property will be used instead

Class Number

Number of [ITile](#)

Inheritance

System.Object  
Number

Implements

System.IEquatable<[Number](#)>

Inherited Members

System.Object.Equals(System.Object, System.Object)  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [GTH02.Tiles.Core.Tiles](#)

Assembly: GTH02.Tiles.Core.dll

Syntax

```
public class Number : IEquatable<Number>
```

Constructors

**Number(Int32, Int32, Int32)**

Creates new [Number](#)

Declaration

```
public Number(int x, int y, int z)
```

Parameters

Type	Name	Description
System.Int32 x	<a href="#">X</a>	
		Should be >= 0
System.Int32 y	<a href="#">Y</a>	
		Should be >= 0
System.Int32 z	Zoom	
		Should be >= 0

Exceptions

Type	Condition
System.ArgumentOutOfRangeException	

Properties

X

X [Number](#) value

Declaration

```
public int X { get; }
```

Property Value

Type	Description
System.Int32	

Y

Y [Number](#) value

Declaration

```
public int Y { get; }
```

Property Value

Type	Description
System.Int32	

Z

Zoom

Declaration

```
public int Z { get; }
```

Property Value

Type	Description
System.Int32	

Methods

Add(Number)

Sum [Numbers](#)

Sums [X](#) and [Y](#) only if [Z](#)s are the same; returns null otherwise

Declaration

```
public Number Add(Number other)
```

Parameters

Type	Name	Description
<a href="#">Number</a>	other	<a href="#">Number</a> to add

Returns

Type	Description
<a href="#">Number</a>	New <a href="#">Number</a> , if <a href="#">Z</a> s are the same

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentException	

Divide(Number)

Divide [Numbers](#)

Divide [X](#) and [Y](#) only if [Z](#)s are the same; returns null otherwise

Declaration

```
public Number Divide(Number other)
```

Parameters

Type	Name	Description
<a href="#">Number</a>	other	<a href="#">Number</a> to divide on

Returns

Type	Description
<a href="#">Number</a>	New <a href="#">Number</a> , if <a href="#">Z</a> s are the same

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentException	

Equals(Number)

Declaration

```
public bool Equals(Number other)
```

Parameters

Type	Name	Description
<a href="#">Number</a>	other	

Returns

Type	Description
System.Boolean	

Equals(Object)

Declaration

```
public override bool Equals(object number)
```

Parameters

Type	Name	Description
System.Object	number	

Returns

Type	Description
System.Boolean	



Overrides

System.Object.Equals(System.Object)

Flip()

Flips [Number](#)

Declaration

```
public Number Flip()
```

Returns

Type	Description
------	-------------

[Number](#) Converted [Number](#)

Flip(Number)

Flips [Number](#)

Declaration

```
public static Number Flip(Number number)
```

Parameters

Type	Name	Description
------	------	-------------

[Number](#) number [Number](#) to flip

Returns

Type	Description
------	-------------

[Number](#) Converted [Number](#)

Exceptions

Type	Condition
System.ArgumentNullException	

GetCount(GeoCoordinate, GeoCoordinate, Int32, Int32, Boolean, Size)

Get count of [Tiles](#) in specified region

Declaration

```
public static int GetCount(GeoCoordinate minCoordinate, GeoCoordinate maxCoordinate, int minZ, int maxZ, bool tmsCompatible, Size tileSize)
```

Parameters

Type	Name	Description
<a href="#">GeoCoordinate</a>	minCoordinate	Minimal <a href="#">GeoCoordinate</a>

[GeoCoordinate](#) maxCoordinate Maximal [GeoCoordinate](#)

System.Int32 minZ Minimal zoom

System.Int32 maxZ Maximal zoom

System.Boolean tmsCompatible Is tms compatible?

[Size](#) tileSize [Tile's Size](#)

Returns

Type	Description
------	-------------

System.Int32 [Tiles](#) count

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentOutOfRangeException	

GetHashCode()

Declaration

```
public override int GetHashCode()
```

Returns

Type	Description
------	-------------

System.Int32

Overrides

System.Object.GetHashCode()

GetLowerNumbers()

Gets 4 one zoom lower [Numbers](#)

Declaration

```
public Number[] GetLowerNumbers()
```

Returns

Type	Description
------	-------------

[Number](#)[] 4 lower [Numbers](#)

Exceptions

Type	Condition
System.ArgumentNullException	

GetLowerNumbers(Number)

Gets 4 one zoom lower [Numbers](#)

Declaration

```
public static Number[] GetLowerNumbers(Number number)
```

Parameters

Type	Name	Description
------	------	-------------

[Number](#) number Input [Number](#)

Returns

Type	Description
------	-------------

[Number](#)[] 4 lower [Numbers](#)

Exceptions

Type	Condition
System.ArgumentNullException	

GetLowerNumbers(Number, Int32)

Get lower [Numbers](#) for specified [Number](#) and zoom

Declaration

```
public static (Number minNumber, Number maxNumber) GetLowerNumbers(Number number, int z)
```

Parameters

Type	Name	Description
------	------	-------------

[Number](#) number Base [Number](#)

System.Int32 z

Returns

Type	Description
System.ValueTuple< <a href="#">Number</a> , <a href="#">Number</a> >	System.ValueTuple`2 of lower <a href="#">Numbers</a>

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentOutOfRangeException	

GetLowerNumbers(Int32)

Get lower [Numbers](#) for specified [Number](#) and zoom

Declaration

```
public (Number minNumber, Number maxNumber) GetLowerNumbers(int z)
```

Parameters

Type	Name	Description
------	------	-------------

System.Int32 z

Zoom  
Must be >= 10

Returns

Type	Description
------	-------------

Type	Description
System.ValueTuple< <a href="#">Number</a> , <a href="#">Number</a> >	System.ValueTuple`2 of lower <a href="#">Numbers</a>

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentOutOfRangeException	

Multiply([Number](#))

Multiply [Numbers](#)

Multiply [X](#) and [Y](#) only if [Z](#)s are the same; returns null otherwise

Declaration

```
public Number Multiply(Number other)
```

Parameters

Type	Name	Description
<a href="#">Number</a>	other	<a href="#">Number</a> to multiply

Returns

Type	Description
<a href="#">Number</a>	New <a href="#">Number</a> , if <a href="#">Z</a> s are the same

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentException	

Subtract([Number](#))

Subtract [Numbers](#)

Subtract [X](#) and [Y](#) only if [Z](#)s are the same; returns null otherwise

Declaration

```
public Number Subtract(Number other)
```

Parameters

Type	Name	Description
<a href="#">Number</a>	other	<a href="#">Number</a> to subtract

Returns

Type	Description
<a href="#">Number</a>	New <a href="#">Number</a> , if <a href="#">Z</a> s are the same

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentException	

ToGeoCoordinates([CoordinateSystem](#), [Size](#), [Boolean](#))

Convert [Number](#) to [GeoCoordinates](#)

Declaration

```
public (GeoCoordinate minCoordinate, GeoCoordinate maxCoordinate) ToGeoCoordinates(CoordinateSystem coordinateSystem, Size tileSize, bool tmsCompatible)
```

Parameters

Type	Name	Description
<a href="#">CoordinateSystem</a>	coordinateSystem	Desired number's coordinate system
<a href="#">Size</a>	tileSize	<a href="#">Tile's Size</a>
System.Boolean	tmsCompatible	Is tms compatible?

Returns

Type	Description
System.ValueTuple< <a href="#">GeoCoordinate</a> , <a href="#">GeoCoordinate</a> >	System.ValueTuple`2 of <a href="#">GeoCoordinates</a>

Exceptions

Type	Condition
System.ArgumentNullException	

**ToGeoCoordinates(Number, CoordinateSystem, Size, Boolean)**

Convert [Number](#) to [GeoCoordinates](#)

**Declaration**

```
public static (GeoCoordinate minCoordinate, GeoCoordinate maxCoordinate) ToGeoCoordinates(Number number, CoordinateSystem coordinateSystem, Size tileSize, bool tmsCompatible)
```

**Parameters**

Type	Name	Description
<a href="#">Number</a>	number	<a href="#">Number</a> to convert
<a href="#">CoordinateSystem</a>	coordinateSystem	
<a href="#">Size</a>	tileSize	
System.Boolean	tmsCompatible	

**Returns**

Type	Description
System.ValueTuple< <a href="#">GeoCoordinate</a> , <a href="#">GeoCoordinate</a> >	System.ValueTuple`2 of <a href="#">GeoCoordinates</a>

**Exceptions**

Type	Condition
System.NotSupportedException	

**ToGeodeticCoordinates(Size, Boolean)**

Convert [Number](#) to [GeodeticCoordinates](#)

**Declaration**

```
public (GeodeticCoordinate minCoordinate, GeodeticCoordinate maxCoordinate) ToGeodeticCoordinates(Size tileSize, bool tmsCompatible)
```

**Parameters**

Type	Name	Description
<a href="#">Size</a>	tileSize	<a href="#">Tile's Size</a>
System.Boolean	tmsCompatible	Is tms compatible?

**Returns**

Type	Description
System.ValueTuple< <a href="#">GeodeticCoordinate</a> , <a href="#">GeodeticCoordinate</a> >	System.ValueTuple`2 of <a href="#">GeodeticCoordinates</a>

**Exceptions**

Type	Condition
System.ArgumentNullException	

**ToGeodeticCoordinates(Number, Size, Boolean)**

Convert [Number](#) to [GeodeticCoordinates](#)

**Declaration**

```
public static (GeodeticCoordinate minCoordinate, GeodeticCoordinate maxCoordinate) ToGeodeticCoordinates(Number number, Size tileSize, bool tmsCompatible)
```

**Parameters**

Type	Name	Description
<a href="#">Number</a>	number	<a href="#">Number</a> to convert
<a href="#">Size</a>	tileSize	
System.Boolean	tmsCompatible	

**Returns**

Type	Description
System.ValueTuple< <a href="#">GeodeticCoordinate</a> , <a href="#">GeodeticCoordinate</a> >	System.ValueTuple`2 of <a href="#">GeodeticCoordinates</a>

**Exceptions**

Type	Condition
System.ArgumentNullException	

**ToMercatorCoordinates(Size, Boolean)**

Convert [Number](#) to [MercatorCoordinates](#)

**Declaration**

```
public (MercatorCoordinate minCoordinate, MercatorCoordinate maxCoordinate) ToMercatorCoordinates(Size tileSize, bool tmsCompatible)
```

Parameters

Type	Name	Description
<a href="#">Size</a>	tileSize	<a href="#">Tile's Size</a>

System.Boolean tmsCompatible Is tms compatible?

Returns

Type	Description
System.ValueTuple< <a href="#">MercatorCoordinate</a> , <a href="#">MercatorCoordinate</a> >	System.ValueTuple`2 of <a href="#">MercatorCoordinates</a>

Exceptions

Type	Condition
System.ArgumentNullException	

ToMercatorCoordinates(Number, Size, Boolean)

Convert [Number](#) to [MercatorCoordinates](#)

Declaration

```
public static (MercatorCoordinate minCoordinate, MercatorCoordinate maxCoordinate) ToMercatorCoordinates(Number number, Size tileSize, bool tmsCompatible)
```

Parameters

Type	Name	Description
<a href="#">Number</a>	number	<a href="#">Number</a> to convert
<a href="#">Size</a>	tileSize	
System.Boolean tmsCompatible		

Returns

Type	Description
System.ValueTuple< <a href="#">MercatorCoordinate</a> , <a href="#">MercatorCoordinate</a> >	System.ValueTuple`2 of <a href="#">MercatorCoordinates</a>

Exceptions

Type	Condition
System.ArgumentNullException	

Operators

Addition(Number, Number)

Sum [Numbers](#)

Sums [X](#) and [Y](#) only if [Z](#)s are the same; returns null otherwise

Declaration

```
public static Number operator +(Number number1, Number number2)
```

Parameters

Type	Name	Description
<a href="#">Number</a>	number1	<a href="#">Number</a> 1
<a href="#">Number</a>	number2	<a href="#">Number</a> 2

Returns

Type	Description
<a href="#">Number</a>	New <a href="#">Number</a> , if <a href="#">Z</a> s are the same

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentException	

Division(Number, Number)

Divide [Numbers](#)

Divide [X](#) and [Y](#) only if [Z](#)s are the same; returns null otherwise

Declaration

```
public static Number operator /(Number number1, Number number2)
```

Parameters

Type	Name	Description
------	------	-------------

Type	Name	Description
------	------	-------------

<a href="#">Number</a>	number1	<a href="#">Number</a> 1
------------------------	---------	--------------------------

<a href="#">Number</a>	number2	<a href="#">Number</a> 2
------------------------	---------	--------------------------

Returns

Type	Description
------	-------------

<a href="#">Number</a>	New <a href="#">Number</a> , if <a href="#">Z</a> s are the same
------------------------	--

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentException	

Equality([Number](#), [Number](#))

Check two [Numbers](#) for equality

Declaration

```
public static bool operator ==(Number number1, Number number2)
```

Parameters

Type	Name	Description
------	------	-------------

<a href="#">Number</a>	number1	<a href="#">Number</a> 1
------------------------	---------	--------------------------

<a href="#">Number</a>	number2	<a href="#">Number</a> 2
------------------------	---------	--------------------------

Returns

Type	Description
System.Boolean	true if <a href="#">Numbers</a> are equal; false otherwise

Inequality([Number](#), [Number](#))

Check two [Numbers](#) for non-equality

Declaration

```
public static bool operator !=(Number number1, Number number2)
```

Parameters

Type	Name	Description
------	------	-------------

<a href="#">Number</a>	number1	<a href="#">Number</a> 1
------------------------	---------	--------------------------

<a href="#">Number</a>	number2	<a href="#">Number</a> 2
------------------------	---------	--------------------------

Returns

Type	Description
System.Boolean	true if <a href="#">Numbers</a> are not equal; false otherwise

Multiply([Number](#), [Number](#))

Multiply [Numbers](#)

Multiply [X](#) and [Y](#) only if [Z](#)s are the same; returns null otherwise

Declaration

```
public static Number operator *(Number number1, Number number2)
```

Parameters

Type	Name	Description
------	------	-------------

<a href="#">Number</a>	number1	<a href="#">Number</a> 1
------------------------	---------	--------------------------

<a href="#">Number</a>	number2	<a href="#">Number</a> 2
------------------------	---------	--------------------------

Returns

Type	Description
------	-------------

<a href="#">Number</a>	New <a href="#">Number</a> , if <a href="#">Z</a> s are the same
------------------------	--

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentException	

Subtraction(Number, Number)

Subtract [Numbers](#)

Subtract [X](#) and [Y](#) only if [Z](#)s are the same; returns null otherwise

Declaration

```
public static Number operator -(Number number1, Number number2)
```

Parameters

Type	Name	Description
<a href="#">Number</a>	number1	<a href="#">Number</a> 1
<a href="#">Number</a>	number2	<a href="#">Number</a> 2

Returns

Type	Description
<a href="#">Number</a>	New <a href="#">Number</a> , if <a href="#">Z</a> s are the same

Exceptions

Type	Condition
System.ArgumentNullException	
System.ArgumentException	

Implements

System.IEquatable<T>

Class RasterTile

[Raster Tile](#)

Inheritance

System.Object  
[Tile](#)  
RasterTile

Implements

[ITile](#)  
System.IDisposable  
System.IAsyncDisposable

Inherited Members

[Tile.DefaultSize](#)  
[Tile.IsDisposed](#)  
[Tile.MinCoordinate](#)  
[Tile.MaxCoordinate](#)  
[Tile.Number](#)  
[Tile.Bytes](#)  
[Tile.Size](#)  
[Tile.Path](#)  
[Tile.Extension](#)  
[Tile.TmsCompatible](#)  
[Tile.MinimalBytesCount](#)  
[Tile.Dispose\(\)](#)  
[Tile.Dispose\(Boolean\)](#)  
[Tile.DisposeAsync\(\)](#)  
[Tile.Validate\(Boolean\)](#)  
[Tile.Validate\(ITile, Boolean\)](#)  
[Tile.CalculatePosition\(\)](#)  
[Tile.CalculatePosition\(Number, Boolean\)](#)  
[Tile.GetExtensionString\(\)](#)  
[Tile.GetExtensionString\(TileExtension\)](#)  
[Tile.WriteToFile\(String\)](#)  
[Tile.WriteToFile\(Tile, String\)](#)  
System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [GTile2Tiles.Core.Tiles](#)

Assembly: GTile2Tiles.Core.dll

Syntax

```
public class RasterTile : Tile, ITile, IDisposable, IAsyncDisposable
```

Constructors

**RasterTile(GeoCoordinate, GeoCoordinate, Int32, Size, IEnumerable<Byte>, TileExtension, Boolean, Int32, Interpolation)**

Declaration

```
public RasterTile(GeoCoordinate minCoordinate, GeoCoordinate maxCoordinate, int zoom, Size size = null, IEnumerable<byte> bytes = null, TileExtension extension = TileExtension.Png,
```

Parameters

Type	Name	Description
<a href="#">GeoCoordinate</a>	minCoordinate	
<a href="#">GeoCoordinate</a>	maxCoordinate	
System.Int32	zoom	
<a href="#">Size</a>	size	
System.Collections.Generic.IEnumerable<System.Byte>	bytes	
<a href="#">TileExtension</a>	extension	
System.Boolean	tmsCompatible	
		<a href="#">BandsCount</a>
System.Int32	bandsCount	Must be in range (0, 4];  <a href="#">DefaultBandsCount</a> by default
		<a href="#">Interpolation</a>
<a href="#">Interpolation</a>	interpolation	<a href="#">Lanczos3</a> by default

Exceptions

Type	Condition
System.ArgumentOutOfRangeException	

**RasterTile(Number, CoordinateSystem, Size, IEnumerable<Byte>, TileExtension, Boolean, Int32, Interpolation)**

Declaration

```
public RasterTile(Number number, CoordinateSystem coordinateSystem, Size size = null, IEnumerable<byte> bytes = null, TileExtension extension = TileExtension.Png, bool tmsCompatil
```

Parameters

Type	Name	Description
<a href="#">Number</a>	number	



Type	Name	Description
<a href="#">CoordinateSystem</a>	coordinateSystem	
<a href="#">Size</a>	size	
System.Collections.Generic.IEnumerable<System.Byte>	bytes	
<a href="#">TileExtension</a>	extension	
System.Boolean	tnsCompatible	
		<a href="#">BandsCount</a>
System.Int32	bandsCount	Must be in range (0, 4]; <a href="#">DefaultBandsCount</a> by default
		<a href="#">Interpolation</a>
<a href="#">Interpolation</a>	interpolation	<a href="#">Lanczos3</a> by default

Exceptions

Type	Condition
System.ArgumentOutOfRangeException	

Fields

DefaultBandsCount

Default count of bands

Declaration

```
public const int DefaultBandsCount = 4
```

Field Value

Type	Description
System.Int32	

Properties

BandsCount

Count of bands in [RasterTile](#)

Declaration

```
public int BandsCount { get; }
```

Property Value

Type	Description
System.Int32	

Interpolation

Interpolation of this [RasterTile](#)

Declaration

```
public Interpolation Interpolation { get; }
```

Property Value

Type	Description
<a href="#">Interpolation</a>	

Implements

[ITile](#)  
System.IDisposable  
System.IAsyncDisposable

Class Tile

Basic implementation of [ITile](#) interface

Inheritance

System.Object  
Tile  
[RasterTile](#)

Implements

[ITile](#)  
System.IDisposable  
System.IAsyncDisposable

Inherited Members

System.Object.Equals(System.Object)  
System.Object.Equals(System.Object, System.Object)  
System.Object.GetHashCode()  
System.Object.GetType()  
System.Object.MemberwiseClone()  
System.Object.ReferenceEquals(System.Object, System.Object)  
System.Object.ToString()

Namespace: [GHI02.Tiles.Core.Tiles](#)

Assembly: GHI02.Tiles.Core.dll

Syntax

```
public class Tile : ITile, IDisposable, IAsyncDisposable
```

Constructors

**Tile(GeoCoordinate, GeoCoordinate, Int32, Size, IEnumerable<Byte>, TileExtension, Boolean)**

Creates new [Tile](#) from [GeoCoordinate](#) values

Declaration

```
protected Tile(GeoCoordinate minCoordinate, GeoCoordinate maxCoordinate, int zoom, Size size = null, IEnumerable<byte> bytes = null, TileExtension extension = TileExtension.Png, bool tmsCompatible = false)
```

Parameters

Type	Name	Description
<a href="#">GeoCoordinate</a>	minCoordinate	Minimal <a href="#">GeoCoordinate</a>
<a href="#">GeoCoordinate</a>	maxCoordinate	Maximal <a href="#">GeoCoordinate</a>
System.Int32	zoom	Zoom
<a href="#">Size</a>	size	<a href="#">Size</a> ; should be a square, e.g. 256x256; If set to null, uses <a href="#">DefaultSize</a> as value
System.Collections.Generic.IEnumerable<System.Byte> bytes	bytes	<a href="#">Bytes</a>
<a href="#">TileExtension</a>	extension	<a href="#">Extension</a>
System.Boolean	tmsCompatible	Is tms compatible?

Exceptions

Type	Condition
System.ArgumentException	

**Tile(Number, CoordinateSystem, Size, IEnumerable<Byte>, TileExtension, Boolean)**

Creates new [Tile](#)

Declaration

```
protected Tile(Number number, CoordinateSystem coordinateSystem, Size size = null, IEnumerable<byte> bytes = null, TileExtension extension = TileExtension.Png, bool tmsCompatible = false)
```

Parameters

Type	Name	Description
<a href="#">Number</a>	number	<a href="#">Number</a>
<a href="#">CoordinateSystem</a>	coordinateSystem	Desired coordinate system
<a href="#">Size</a>	size	<a href="#">Size</a> ; should be a square, e.g. 256x256; If set to null, uses <a href="#">DefaultSize</a> as value
System.Collections.Generic.IEnumerable<System.Byte> bytes	bytes	<a href="#">Bytes</a>
<a href="#">TileExtension</a>	extension	<a href="#">Extension</a>
System.Boolean	tmsCompatible	Is tms compatible?

Exceptions

Type	Condition
System.ArgumentException	

Fields

DefaultSize

Default [Tile's Size](#)

Uses Gtiff2Tiles.Core.Tiles.Tile.DefaultSideSizeValue as values for width and height

Declaration

```
public static readonly Size DefaultSize
```

Field Value

Type Description

[Size](#)

Properties

Bytes

Collection of [ITile](#)'s bytes

Declaration

```
public IEnumerable<byte> Bytes { get; set; }
```

Property Value

Type	Description
System.Collections.Generic.IEnumerable<System.Byte>	

Extension

Extension of [ITile](#)

Declaration

```
public TileExtension Extension { get; }
```

Property Value

Type	Description
<a href="#">TileExtension</a>	

IsDisposed

Shows if this [ITile](#)'s already disposed

Declaration

```
public bool IsDisposed { get; }
```

Property Value

Type	Description
System.Boolean	

MaxCoordinate

Maximal [GeoCoordinate](#) of this [ITile](#)

Declaration

```
public GeoCoordinate MaxCoordinate { get; }
```

Property Value

Type	Description
<a href="#">GeoCoordinate</a>	

MinCoordinate

Minimal [GeoCoordinate](#) of this [ITile](#)

Declaration

```
public GeoCoordinate MinCoordinate { get; }
```

Property Value

Type	Description
<a href="#">GeoCoordinate</a>	

MinimalBytesCount

355 by default

Declaration

```
public int MinimalBytesCount { get; set; }
```

Property Value

Type	Description
System.Int32	

Number

[Number](#) of this [ITile](#)

Declaration

```
public Number Number { get; }
```

Property Value

Type	Description
<a href="#">Number</a>	

Path

Path on disk of this [ITile](#)

Declaration

```
public string Path { get; set; }
```

Property Value

Type	Description
System.String	

Size

[Size](#) (width and height) of this [ITile](#)

Declaration

```
public Size Size { get; }
```

Property Value

Type	Description
<a href="#">Size</a>	

TmsCompatible

Is [ITile](#) tms compatible?

Declaration

```
public bool TmsCompatible { get; }
```

Property Value

Type	Description
System.Boolean	

Methods

CalculatePosition()

Calculates this [ITile](#)'s position in upper [ITile](#)

Declaration

```
public int CalculatePosition()
```

Returns

Type	Description
	Value in range from 0 to 3
System.Int32	Starts always from upper-left corner and goes to lower-right, but maths depends on <a href="#">TmsCompatible</a> value

CalculatePosition(Number, Boolean)

Calculates this [ITile](#)'s position in upper [ITile](#)

Declaration

```
public static int CalculatePosition(Number number, bool tmsCompatible)
```

Parameters

Type	Name	Description
<a href="#">Number</a>	number	<a href="#">Number</a> of <a href="#">Tile</a>
System.Boolean	tmsCompatible	Is tms compatible?

Returns

Type	Description
	Value in range from 0 to 3
System.Int32	Starts always from upper-left corner and goes to lower-right, but maths depends on <a href="#">TmsCompatible</a> value

Exceptions

Type	Condition
System.ArgumentNullException	

Dispose()

Declaration

```
public void Dispose()
```

**Dispose(Boolean)**

**Declaration**

```
protected virtual void Dispose(bool disposing)
```

**Parameters**

Type	Name	Description
System.Boolean	disposing	Dispose static fields?

**DisposeAsync()**

**Declaration**

```
public ValueTask DisposeAsync()
```

**Returns**

Type	Description
System.Threading.Tasks.ValueTask	

**Finalize()**

Calls [Dispose\(Boolean\)](#) on this [Tile](#)

**Declaration**

```
protected void Finalize()
```

**GetExtensionString()**

Get System.String from [TileExtension](#)

**Declaration**

```
public string GetExtensionString()
```

**Returns**

Type	Description
System.String	Converted System.String

**GetExtensionString(TileExtension)**

Get System.String from [TileExtension](#)

**Declaration**

```
public static string GetExtensionString(TileExtension extension)
```

**Parameters**

Type	Name	Description
<a href="#">TileExtension</a>	extension	<a href="#">TileExtension</a> to convert

**Returns**

Type	Description
System.String	Converted System.String

**Exceptions**

Type	Condition
System.ArgumentOutOfRangeException	

**Validate(ITile, Boolean)**

Checks if this [ITile](#) is not empty or too small  
See [MinimalBytesCount](#) property for more info

**Declaration**

```
public static bool Validate(ITile tile, bool isCheckPath)
```

**Parameters**

Type	Name	Description
<a href="#">ITile</a>	tile	<a href="#">Tile</a> to check
System.Boolean	isCheckPath	

**Returns**

Type	Description
System.Boolean	true if <a href="#">ITile</a> 's valid; false otherwise

**Validate(Boolean)**

Checks if this [ITile](#) is not empty or too small

See [MinimalBytesCount](#) property for more info

Declaration

```
public bool Validate(bool isCheckPath)
```

Parameters

Type	Name	Description
System.Boolean	isCheckPath	Do you want to check <a href="#">Path</a> ?

Returns

Type	Description
System.Boolean	true if <a href="#">ITile</a> 's valid; false otherwise

WriteToFile(ITile, String)

Declaration

```
public static void WriteToFile(ITile tile, string path = null)
```

Parameters

Type	Name	Description
<a href="#">ITile</a>	tile	<a href="#">ITile</a> to write
System.String	path	

Exceptions

Type	Condition
System.ArgumentNullException	

WriteToFile(String)

Writes [ITile](#)'s [Bytes](#) to file

Declaration

```
public void WriteToFile(string path = null)
```

Parameters

Type	Name	Description
System.String	path	Full path to write <a href="#">ITile</a> if not set, <a href="#">Path</a> property will be used instead

Exceptions

Type	Condition
System.ArgumentNullException	

Implements

[ITile](#)  
System.IDisposable  
System.IAsyncDisposable