

Table of Contents

Articles

[Welcome](#)

[License Agreement](#)

[Getting Started](#)

[Copyright](#)

[Version History](#)

Api Documentation

[BitMiracle.Jpeg2k](#)

[J2kCodec](#)

[J2kColorSpace](#)

[J2kDecodingOptions](#)

[J2kEncodingOptions](#)

[J2kException](#)

[J2kImage](#)

[J2kImageComponent](#)

[J2kImageComponentInfo](#)

[J2kImageComponentPrecision](#)

[J2kImageData](#)

[J2kImportOptions](#)

[J2kOpenOptions](#)

[J2kOutputFormat](#)

[J2kPixels](#)

[J2kPrecisionMode](#)

[J2kProgressionOrder](#)

[J2kQualityMode](#)

[J2kTileCodingStyle](#)

[J2kTileInfo](#)

[LicenseManager](#)

Jpeg2000.Net

Jpeg2000.Net is the royalty-free JPEG 2000 codec available as a pure .NET library.

The codec is written in C# without unsafe code and external dependencies.

The codec is available for WinForms, WPF and ASP.NET. A version for .NET Standard Library / .NET Core / ASP.NET Core is also available.

Features

- Can compress and decompress JPEG 2000 images
- Lossy or lossless compression
- Alpha-channel support
- Can decode and encode images with up to 16 bits per component
- Can decode only part of an image
- Can decode particular tile only
- Can limit number of quality layers to be decoded
- Allows different compression ratios for different layers
- Allows to choose number of resolutions in the encoded image
- 100% managed, without unsafe blocks
- Available for .NET Standard Library / .NET Core / ASP.NET Core
- No external dependencies

BIT MIRACLE END USER LICENSE AGREEMENT

JPEG2000.NET LICENSE

IMPORTANT: READ CAREFULLY: This End User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and Bit Miracle for the materials accompanying this EULA, which may include computer software, associated media, printed materials, and "on line" or electronic documentation ("Software") and source files, resource files, project and solution files ("Source Code") for Software, collectively "The Product".

BY INSTALLING, DOWNLOADING, COPYING OR OTHERWISE USING THE PRODUCT, YOU AGREE TO BE BOUND BY THE TERMS OF THIS EULA. IF YOU DO NOT AGREE TO THE TERMS OF THIS EULA, DO NOT INSTALL, DOWNLOAD, COPY OR USE THE PRODUCT.

The Product is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. Bit Miracle or its suppliers own the title, copyright and other intellectual property rights in The Product. The Product is licensed, not sold.

1. DEFINITIONS.

- a. "Application Programming Interface" or "API" means a publicly accessible interface defining the ways by which an application program may request services from libraries and/or software.
- b. "Software as a Service" or "SaaS" means a model of software deployment whereby a provider licenses an application to customers for use as a service on demand.
- c. "Server product" means the software of a type that is designed to function as a server (including SaaS) and thereby provide data over the internet or any kind of network or multi-user environment.
- d. "End User Application" means the software of a type that is designed to be used on a workstation computer by a final user to perform a function or type of work.
- e. "Software Development Kit" or "SDK" means a set of development tools that allows a software engineer to create applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar platform.
- f. "You" are the person or organization licensing The Product and are accordingly, the Licensee in respect of it.
- g. "Link to" means The Product is linked in binary form to a software product and is required by the build process of that software product to build successfully.
- h. Source Code is the mnemonic, high-level statement versions of The Product written in the source language used by programmers.

2. GRANT OF LICENSE. Bit Miracle grants you the rights described in this EULA provided that you comply with all the terms and conditions of this EULA:

- a. General Software License Grant. Bit Miracle grants use of The Product according to one of the license types below as identified in The Product title.
- i. Evaluation License. The Evaluation License is a non-exclusive evaluation license to use The Product for evaluation purposes only. With the Evaluation License The Product is fully functional for a limited number of days. The exact number of days is to be decided by Bit Miracle in for each Evaluation License. When using the Evaluation License The Product may be installed for internal evaluation purposes only and MAY NOT be used outside of a development environment. Any violation of this provision shall require a mandatory purchase of any required license, as well as expose the user to other legal recourse for collection and punitive damages. If you do not agree to these terms, do not evaluate The Product and remove it from your computer immediately.
- ii. Ultimate License. This Licenses grants You non-transferable and non-exclusive right to link The Product to End User

Applications, Server Products and Cloud Solutions developed, marketed and distributed by You and distributed either within your organization or to third parties.

b. General Source Code License Grant. If Source Code accompanies The Product, Bit Miracle grants to you one (1) personal, non-transferable, non-exclusive, royalty-free license to make and use copies of the Source Code and install such Source Code on any number of your computers (i) for your internal use, (ii) to design, develop and test your software products. You may not redistribute the Source Code, or any component thereof, whether modified or not to any third party.

c. Documentation. With respect to electronic and other documentation, you may make any number of copies (either in hard copy or electronic form) provided that such copies shall be used only for internal purposes and are not republished or distributed externally.

d. Disassembly. You may not reverse engineer, decompile, disassemble or in any other way try to gain access to information regarding the construction of The Product.

e. Reservation of Rights. Bit Miracle reserves all rights not expressly granted herein.

3. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS.

a. Derived Works and Redistribution

i. The Product may not be distributed in any form that allows it to be reused by any application other than your derived work. The Product may not be included as part of an SDK. The Product may not be used to develop derived works that offer similar functionality as The Product, or expose features of The Product through an API for use by an unlicensed third party.

ii. The Product may not be used to develop derived works that offer similar functionality as The Product for any development platform, including but not limited to: .NET, .NET Core, .NET Standard, Microsoft Azure, Mono or MonoTouch, Xamarin.

iii. Bit Miracle is not obligated to provide support for works derived from The Product.

iv. Distribution of the Source Code in any form, with or without modifications, is strictly prohibited.

v. Works derived from The Product may not be called "Bit Miracle" and/or "Jpeg2000.Net", nor may "Bit Miracle" and/or "Jpeg2000.Net" appear in their names, without prior written permission from Bit Miracle.

vi. The names "Bit Miracle" and/or "Jpeg2000.Net" must not be used to endorse or promote products derived from The Product without prior written permission. For written permission, please contact support@bitmiracle.com.

b. Rental. You may not rent, lease, or lend The Product without express written permission from Bit Miracle.

c. Termination. Without prejudice to any other rights, Bit Miracle may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of The Product, including but not limited to the Source Code, backups and all of its component parts and derived works.

d. Consideration. For the rights and license granted in this EULA, including both the General Software License and the General Source Code License, you will pay Bit Miracle the currently published price available at bitmiracle.com or another mutually agreed upon amount to appear on a valid invoice.

e. Term. The term of this EULA shall continue perpetually from the date of purchase unless terminated according to the provisions in Section 3(c).

f. Consent to Use of Data. You agree that Bit Miracle and its affiliates may collect and use technical information, excluding any confidential information, gathered as part of The Product support services provided to you, if any, related to The Product. Bit Miracle may use this information solely to improve our products or to provide customized services or technologies to you. Bit Miracle may disclose this information, excluding any information explicitly declared as confidential information, to others but not in a form that personally identifies you.

4. SOFTWARE MAINTENANCE, UPDATES AND DISCONTINUATION.

- a. A purchase of The Product entitles the purchaser to updates of The Product for a period of one (1) year (365 days) following the purchase.
- b. Bit Miracle may, in its sole discretion, provide technical support, updates and/or supplements of the Software and/or related information ("Updates") to You hereunder, in which case such Updates shall also be deemed to be included in The Product and therefore governed by this EULA, unless other terms of use are provided by Bit Miracle with such Updates.
- c. Bit Miracle reserves the right to discontinue The Product at any time, whether it is offered individually or as a part of a product suite.
- d. If for any reason Bit Miracle decides to discontinue The Product they will provide the following remedy to each license owner:
 - i. Bit Miracle will make a reasonable effort to notify you in writing at least fifteen (15) days prior to the discontinuation of The Product, and you will be provided with The Product Source Code.
- e. If for any reason, including insolvency or dissolution, Bit Miracle is unable to remain in business they will provide the following remedy to each license owner:
 - i. Bit Miracle will make a reasonable effort to notify you at least thirty (30) days prior to close of business, and you will be provided with The Product Source Code for The Product you own licenses for.

5. SOFTWARE DELIVERY. The Product and any associated materials are provided in electronic format only. You are responsible for downloading The Product from the Bit Miracle website: <https://bitmiracle.com>. Upon purchase Bit Miracle shall deliver to you an electronic license key which will enable The Product to function in the purchased license capacity.

6. UPGRADES. If this copy of The Product is an upgrade from an earlier version of The Product, it is provided to you on a license exchange basis. You agree by your installation and use of such copy of The Product to voluntarily terminate your earlier EULA and that you will not continue to use the earlier version of The Product or transfer it to another person or entity unless such transfer is otherwise valid according to the terms of this EULA.

7. INTELLECTUAL PROPERTY RIGHTS. All title and intellectual property rights in and to The Product (including but not limited to any images, photographs, animations, video, audio, music and text incorporated into The Product and any copies of The Product that you are expressly permitted to make herein) are owned by Bit Miracle or its suppliers. All title and intellectual property rights in and to the content which may be accessed through use of The Product are the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This EULA grants you no rights to use such content. All title and intellectual property rights in and to your modifications to The Product, are owned by you provided that such modifications are made available to Bit Miracle free of charge and royalty-free for inclusion in later releases of The Product. All rights not expressly granted are reserved by Bit Miracle.

8. NONDISCLOSE AND NONCOMPETITION.

- a. Both parties recognize that the other party may obtain proprietary and/or confidential information ("Confidential Information") in the course of the business relationship defined herein. Each party therefore agrees that it will not disclose the Confidential Information of the other party to any third party (i) except as provided for in this EULA or as necessary to pursue and implement the business relationships defined herein; (ii) to its responsible employees and professional advisors with a bona fide need to know and whom are bound by agreement or law to keep such information confidential; (iii) as authorized by the other party in writing or (iv) to the extent required by applicable law, court, or government agency, provided that the receiving party promptly notifies the disclosing party thereof and cooperates with any efforts by the disclosing party, at the disclosing party's expense, to limit such disclosure by means of seeking a protective order or requesting confidential treatment.
- b. The parties agree that any party that violates this Section 8 shall pay, liquidated damages of no less than four (4) times the purchase price or five thousand dollars (\$5,000.00 USD) whichever is more if any violation of this Section 8 is proved in a court of law or admitted plus any direct, indirect, or actual damages.

9. LIMITED WARRANTY AND DISCLAIMER.

a. Except with respect to an evaluation version of The Product, Bit Miracle warrants that, for a period of thirty (30) days from the date of delivery (as evidenced by a copy of your receipt) when used with a recommended hardware configuration, The Product will perform in substantial conformance with the documentation supplied with The Product.

b. BIT MIRACLE PROVIDES NO REMEDIES OR WARRANTIES, WHETHER EXPRESS OR IMPLIED, FOR EVALUATION VERSIONS OF THE PRODUCT. THE EVALUATION VERSION OF THE PRODUCT IS PROVIDED "AS IS".

c. BIT MIRACLE AND ITS SUPPLIERS DISCLAIM ALL OTHER WARRANTIES AND REPRESENTATIONS, WHETHER EXPRESS, IMPLIED, OR OTHERWISE, INCLUDING THE WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. BIT MIRACLE DOES NOT WARRANT THAT THE PRODUCT IS ERROR-FREE OR WILL OPERATE WITHOUT INTERRUPTION. THE PRODUCT IS NOT DESIGNED, INTENDED OR LICENSED FOR USE IN HAZARDOUS ENVIRONMENTS REQUIRING FAIL-SAFE CONTROLS, INCLUDING WITHOUT LIMITATION, THE DESIGN, CONSTRUCTION, MAINTENANCE OR OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL, AND LIFE SUPPORT OR WEAPONS SYSTEMS. BIT MIRACLE SPECIFICALLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR SUCH PURPOSES.

d. IF APPLICABLE LAW REQUIRES ANY WARRANTIES WITH RESPECT TO THE PRODUCT, ALL SUCH WARRANTIES ARE LIMITED IN DURATION TO NINETY (90) DAYS FROM THE DATE OF DELIVERY.

e. NO ORAL OR WRITTEN INFORMATION OR ADVICE GIVEN BY BIT MIRACLE, ITS DEALERS, DISTRIBUTORS, AGENTS OR EMPLOYEES SHALL CREATE A WARRANTY OR IN ANY WAY INCREASE THE SCOPE OF ANY WARRANTY PROVIDED HEREIN.

10. EXCLUSIVE REMEDY. Your exclusive remedy under the preceding is to return The Product to the place you acquired it, with a copy of your receipt and a description of the problem. Provided that any non-compliance with the above warranty is reported in writing to Bit Miracle no more than thirty (30) days following delivery to you, Bit Miracle will use reasonable commercial efforts to supply you with a replacement copy of The Product that substantially conforms to the documentation or refund to you your purchase price for The Product, at its option. Bit Miracle shall have no responsibility if The Product has been altered in any way or if the failure arises out of use of The Product with other than a recommended hardware configuration. Any such misuse, accident, abuse, modification or misapplication of The Product will void the warranty above. THIS REMEDY IS THE SOLE AND EXCLUSIVE REMEDY AVAILABLE TO YOU FOR BREACH OF EXPRESS OR IMPLIED WARRANTIES WITH RESPECT TO THE PRODUCT.

11. LIMITATION OF LIABILITY.

a. Except with regard to: (i) breach of confidentiality obligations; and (ii) any costs associated with the defense, chosen by Bit Miracle, and settlement, to the infringed upon party, of an intellectual property right claim.

b. NEITHER BIT MIRACLE NOR ITS SUPPLIERS SHALL BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, COVER OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, DAMAGES FOR THE INABILITY TO USE EQUIPMENT OR ACCESS DATA, LOSS OF BUSINESS, LOSS OF PROFITS, BUSINESS INTERRUPTION OR THE LIKE), ARISING OUT OF THE USE OF, OR INABILITY TO USE, THE PRODUCT AND BASED ON ANY THEORY OF LIABILITY INCLUDING BREACH OF CONTRACT, BREACH OF WARRANTY, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE, EVEN IF BIT MIRACLE OR ITS REPRESENTATIVES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES AND EVEN IF A REMEDY SET FORTH HEREIN IS FOUND TO HAVE FAILED OF ITS ESSENTIAL PURPOSE.

c. BIT MIRACLE'S TOTAL LIABILITY TO YOU FOR ACTUAL DAMAGES, FOR ANY CAUSE WHATSOEVER WILL BE LIMITED TO THE AMOUNT PAID BY YOU FOR THE PRODUCT THAT CAUSED SUCH DAMAGE.

d. THE FOREGOING LIMITATIONS ON LIABILITY ARE INTENDED TO APPLY TO THE WARRANTIES AND DISCLAIMERS ABOVE AND ALL OTHER ASPECTS OF THIS EULA.

12. INDEMNITY. You agree to hold Bit Miracle harmless against, and at your expense handle and defend, any claim and defend

any third party suit brought against you based upon an allegation that any software developed by you and included in derived works produced using The Product constitutes an infringement of any international patent, or any copyright or other proprietary or intellectual property right. You shall pay all damages and costs awarded in such suit. Bit Miracle will indemnify, defend and hold you, and/or your and/or its affiliated and/or subsidiary companies harmless against any claims, legal actions, losses and other expenses arising out of or in connection with any claims that The Product and/or related documentation infringes or violates any intellectual property right of any third party ("Claim"), on the condition that you notify Bit Miracle promptly of the Claim and give Bit Miracle sole control of the defense and negotiations for its settlement or compromise. If you become, or may become, prohibited from continued use of the Software and/or The Product by reason of an actual or anticipated Claim, Bit Miracle will use its reasonable efforts to

a. obtain for you the right to use the Software and/or The Product, or

b. replace or modify such Software and/or The Product so that it is no longer subject to a Claim, but performs the same functions in an equivalent manner.

c. In the event that Bit Miracle determines, in its sole discretion, that neither (a) nor (b) is commercially reasonable, Bit Miracle shall refund the residual value of the license fees paid by you for the infringing Product or Software, depreciated using a straight line method of depreciation over a forty-eight month period from the date of delivery of The Product to you.

13. **TAXES.** The license fees and any other amounts payable pursuant to the terms and conditions herein are exclusive of all national, state, regional, local municipal or other taxes and fees including, but not limited to, excise, sales, use, property, ad valorem, intangibles, goods and services and value added taxes, customers duties and registration fees, now in force or enacted in the future, and all such taxes and fees, except taxes based on Bit Miracle's net worth, capital or net income, shall be paid directly by you, or if paid by Bit Miracle, you will reimburse Bit Miracle.
14. **NO WAIVER.** No action taken by either party pursuant to this EULA, and no waiver by either party, whether express or implied, of any provision or right in this EULA or any breach thereof, and no failure of either party to exercise or enforce any of its rights under this EULA, will constitute a continuing waiver with respect to such provision or right or as a breach or waiver or any other provision or right, whether or not similar.
15. **SEVERABILITY.** If any covenant or provision of the EULA is determined to be void or unenforceable in whole or part, then such void or unenforceable covenant or provision shall be deleted from this EULA and shall not effect or impair the enforceability or validity of any other covenant or provision of this EULA or any part thereof.
16. **GOVERNING LAW.** This Agreement shall be governed by and construed in accordance with the laws of the Russian Federation. Any dispute, controversy or claim arising out of this Agreement or relating to it, that cannot be resolved through discussion and consultation among the representatives of the parties, shall be referred to the final settlement to The International Commercial Arbitration Court at the RF Chamber of Commerce and Industry in Moscow, Russian Federation, in conformity with the Rules of the above Court. The arbitration shall be conducted in the Russian language.
17. **FURTHER INSTRUMENTS.** Except as otherwise expressly provided in this Agreement, each party shall furnish to the other (and shall deliver and cause to be executed, acknowledged and delivered to the other) any further instruments, which such other party may reasonably require or deem necessary from time to time to evidence, establish, protect, enforce, defend or secure to such other party any or all of its rights hereunder or to more effectuate or carry out the purposes, provisions or intent of this Agreement.
18. **CAPTIONS.** All indexes, titles, subject headings, section titles, and similar items are provide for the purpose of reference and convenience and are not intended to be inclusive, definitive, or to affect the meaning or scope of this Agreement.
19. **UPDATES.** Bit Miracle may make updates and changes to this license agreement from time to time. Such changes will be posted on our website at <https://bitmiracle.com>. Such changes or updates will become effective immediately. It is your responsibility to check the website for updates. If this EULA is provided with The Product installer, please visit our website to ensure have the most recent agreement. If you elect not to be bound by the new license agreement you must immediately destroy all copies of The Product and inform Bit Miracle in writing within sixty (60) days of posting.

20. ENTIRE AGREEMENT. This EULA is the entire agreement between you and Bit Miracle relating to The Product and the support services (if any) and it supersedes all prior or contemporaneous oral or written communications, proposals and representations with respect to The Product or any other subject matter covered by this EULA. To the extent the terms of any Bit Miracle policies or programs for support services conflict with the terms of this EULA, the terms of this EULA shall control.

Installing the Jpeg2000.Net library

Installing from NuGet

The easiest way to get started is to install the BitMiracle.Jpeg2k package from NuGet.

Installing manually

1. Download the latest release of the Jpeg2000.Net library from our site. The library is distributed as a ZIP package.

The ZIP package contains two versions of the library:

- version for .NET 4.0 and later frameworks
- version for .NET Standard 1.3 and later frameworks

The ZIP package also contains help file, all sample projects, and the file with the license agreement.

2. Extract the downloaded ZIP package to a location of your choice.

Running samples

Jpeg2000.Net samples are located in the Samples folder of the ZIP package. Open SamplesCSharp solution file if you want to use sample code written in C# language. For a VB.NET version please open the SamplesVB.NET solution file.

The same sample code can be cloned or downloaded from our samples repository on GitHub.

Please take a time to review the samples. It should help you to add JPEG 2000 processing features to your application.

Jpeg2000.Net

(C) 2018-2019 Bit Miracle

All rights reserved.

Warning: This computer program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.

This software is based in part on OpenJPEG

<http://www.openjpeg.org/>

All rights reserved.

The copyright in this software is being made available under the 2-clauses BSD License, included below. This software may be subject to other third party and contributor rights, including patent rights, and no such rights are granted under this license.

Copyright (c) 2002-2014, Universite catholique de Louvain (UCL), Belgium Copyright (c) 2002-2014, Professor Benoit Macq
Copyright (c) 2003-2014, Antonin Descampe Copyright (c) 2003-2009, Francois-Olivier Devaux Copyright (c) 2005, Herve Drolon,
Freemage Team Copyright (c) 2002-2003, Yannick Verschuere Copyright (c) 2001-2003, David Janssens Copyright (c) 2011-
2012, Centre National d'Etudes Spatiales (CNES), France Copyright (c) 2012, CS Systemes d'Information, France

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS 'AS IS' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Version 2.1

- Implemented support for multi-threading decoding. Take a look at the new `J2kDecodingOptions.ThreadCount` property
- Optimized speed of Jpeg2000 decoding
- Fixed bugs related to encoding and decoding of JPEG 2000 images

Version 2.0

March 10, 2019

- Greatly optimized speed and memory consumption in `J2kImage.DecodeArea()` method
- Optimized memory consumption for Jpeg2000 decoding
- Optimized speed of Jpeg2000 encoding and decoding
- Fixed bugs related to encoding and decoding of JPEG 2000 images
- Fixed bugs in `J2kImage.DecodeArea()` method

Version 1.0

May 1st, 2018

- Initial release

Namespace BitMiracle.Jpeg2k

Classes

[J2kDecodingOptions](#)

The options for JPEG 2000 image decoding.

[J2kEncodingOptions](#)

The options to use when creating a JPEG 2000 image.

[J2kException](#)

Class for Jpeg2000.Net specific exceptions.

[J2kImage](#)

Properties and methods for a JPEG 2000 image.

[J2kImageComponent](#)

Class for a single component of a JPEG 2000 compatible image data.

[J2kImageComponentInfo](#)

Class for a single component of a JPEG 2000 image.

[J2kImageComponentPrecision](#)

Describes how an image component precision should be changed during decoding.

[J2kImageData](#)

Encapsulates information about a JPEG 2000 compatible image data.

[J2kImportOptions](#)

The options to use while creating [J2kImageData](#) from an image in one of the supported image formats.

[J2kOpenOptions](#)

The options to be used while opening a JPEG 2000 image.

[J2kPixels](#)

Provides access to pixels in [J2kImageComponent](#).

[J2kTileInfo](#)

Provides information about tile in a JPEG 2000 image.

[LicenseManager](#)

Class for license management. Provides properties and methods to add license to the Jpeg2000.Net library.

Enums

[J2kCodec](#)

Enumerations of all supported JPEG 2000 codecs.

[J2kColorSpace](#)

Supported JPEG 2000 image color spaces

[J2kOutputFormat](#)

Enumeration of all possible output formats

[J2kPrecisionMode](#)

Enumeration of all possible image component precision forcing modes.

[J2kProgressionOrder](#)

Enumeration of all possible progression orders.

[J2kQualityMode](#)

Supported encoding quality modes.

[J2kTileCodingStyle](#)

Enumeration of all possible coding styles.

Enum J2kCodec

Enumerations of all supported JPEG 2000 codecs.

Namespace: [BitMiracle.Jpeg2k](#)

Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public enum J2kCodec
```

Fields

| NAME | DESCRIPTION |
|------|-----------------------|
| J2k | JPEG-2000 codestream. |
| Jp2 | JP2 file format. |

Enum J2kColorSpace

Supported JPEG 2000 image color spaces

Namespace: [BitMiracle.Jpeg2k](#)

Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public enum J2kColorSpace
```

Fields

| NAME | DESCRIPTION |
|-------------|----------------------------------|
| Cmyk | CMYK color space. |
| Eycc | e-YCC color space. |
| Gray | Grayscale color space. |
| Srgb | sRGB color space. |
| Sycc | sYCC (YUV) color space. |
| Unknown | Not supported by the library. |
| Unspecified | Not specified in the codestream. |

Class J2kDecodingOptions

The options for JPEG 2000 image decoding.

Inheritance

System.Object
J2kDecodingOptions

Inherited Members

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

Namespace: [BitMiracle.Jpeg2k](#)
Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public class J2kDecodingOptions
```

Properties

ComponentPrecision

Gets or sets information about how to change image components precision.

Declaration

```
public J2kImageComponentPrecision[] ComponentPrecision { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
|--|-------------|
| J2kImageComponentPrecision[] | |

Remarks

Number of elements in this array must be less or equal to the number of image components.

If this property is `null` then components precision is not changed.

ForceRgbColorSpace

Gets or sets a value indicating whether to force output colorspace to RGB.

Declaration

```
public bool ForceRgbColorSpace { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
|----------------|-------------|
| System.Boolean | |

ThreadCount

Gets or sets the number of threads to use for decoding.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Remarks

Default value: 1

The value should be greater than or equal to 1.

UpsampleComponents

Gets or sets a value indicating whether to upsample components.

Declaration

```
public bool UpsampleComponents { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
|----------------|-------------|
| System.Boolean | |

Class J2kEncodingOptions

The options to use when creating a JPEG 2000 image.

Inheritance

System.Object
J2kEncodingOptions

Inherited Members

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

Namespace: [BitMiracle.Jpeg2k](#)
Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public class J2kEncodingOptions
```

Properties

Codec

Gets or sets the codec to use when creating the image.

Declaration

```
public J2kCodec Codec { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------------------|-------------|
| J2kCodec | |

InitialCodeBlockHeight

Gets or sets the initial code block height.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

InitialCodeBlockWidth

Gets or sets the initial code block width.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

InitialPrecintSizes

Gets or sets the initial precinct sizes.

Declaration

```
public int[] InitialPrecintSizes { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
|----------------|-------------|
| System.Int32[] | |

Remarks

The array must contain even number of elements. Each pair of elements in the array specify size of a precinct. The first value in a pair is the width, the second value is the height of the precinct.

Values specified must be power of 2. Multiple pairs may be supplied, in which case the first pair refers to the highest resolution level and subsequent pairs to lower resolution levels. The last specified pair is reused for each remaining lower resolution levels.

When this parameter is `null`, the default initial precinct sizes are used.

Default initial precinct size is $2^{15} \times 2^{15}$. This means 1 precinct.

ProduceTiledImage

Gets or sets a value indicating whether to produce image that consists of more than one tile.

Declaration

```
public bool ProduceTiledImage { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
|----------------|-------------|
| System.Boolean | |

ProgressionOrder

Gets or sets the progression order to use when creating the image.

Declaration

```
public J2kProgressionOrder ProgressionOrder { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
|-------------------------------------|-------------|
| J2kProgressionOrder | |

QualityMode

Gets or sets the value indicating how to treat quality values specified by [QualityValues](#)

Declaration

```
public J2kQualityMode QualityMode { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------------------------|-------------|
| J2kQualityMode | |

QualityValues

Gets or sets the quality values to use while encoding.

Declaration

```
public float[] QualityValues { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
|-----------------|-------------|
| System.Single[] | |

Remarks

Meaning of the quality values depend on the value of the [QualityMode](#).

If the quality mode is `QualityMode.CompressionRatio` then the values of this array are compression ratios for successive layers. The rate specified for each quality level is the desired compression factor. Decreasing ratios required. Values 1 and 0 are treated the same. Example: 20, 10, 1 means

- quality layer 1: compress 20x
- quality layer 2: compress 10x
- quality layer 3: compress lossless

If the quality mode is `QualityMode.FixedQuality` then the values of this array are Peak signal-to-noise ratios (PSNR) for successive layers. Increasing ratios required. Example: 30, 40, 50.

When talking about PSNR in context of image compression the signal is the original data, and the noise is the error introduced by compression. PSNR is an approximation to human perception of reconstruction quality.

Typical values for the PSNR in lossy image and video compression are between 30 and 50 dB, provided the bit depth is 8 bits, where higher is better. For 16-bit data typical values for the PSNR are between 60 and 80 dB.

ResolutionLevelCount

Gets or sets the number of resolution levels in the output image.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

TileHeight

Gets or sets the height of the tiles in the output image.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Remarks

This paramater is ignored if [ProduceTiledImage](#) is `false`.

TileWidth

Gets or sets the width of the tiles in the output image.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Remarks

This paramater is ignored if [ProduceTiledImage](#) is `false`.

Class J2kException

Class for Jpeg2000.Net specific exceptions.

Inheritance

System.Object
System.Exception
J2kException

Implements

System.Runtime.Serialization.ISerializable
System.Runtime.InteropServices._Exception

Inherited Members

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

Namespace: [BitMiracle.Jpeg2k](#)

Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
[Serializable]  
public class J2kException : Exception, ISerializable, _Exception
```

Constructors

J2kException()

Initializes a new instance of the [J2kException](#) class without providing a message.

Declaration

```
public J2kException()
```

J2kException(SerializationInfo, StreamingContext)

Initializes a new instance of the [J2kException](#) class.

Declaration

```
protected J2kException(SerializationInfo info, StreamingContext context)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|--|---------|--|
| System.Runtime.Serialization.SerializationInfo | info | The System.Runtime.Serialization.SerializationInfo that holds the serialized object data about the exception being thrown. |
| System.Runtime.Serialization.StreamingContext | context | The System.Runtime.Serialization.StreamingContext that contains contextual information about the source or destination. |

Exceptions

| TYPE | CONDITION |
|---|---|
| System.ArgumentNullException | The <code>info</code> parameter is null. |
| System.Runtime.Serialization.SerializationException | The class name is null or System.Exception.HResult is zero (0). |

J2kException(String)

Initializes a new instance of the [J2kException](#) class.

Declaration

```
public J2kException(string message)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|---------------|---------|--------------|
| System.String | message | The message. |

J2kException(String, Exception)

Initializes a new instance of the [J2kException](#) class.

Declaration

```
public J2kException(string message, Exception innerException)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|------------------|----------------|----------------------|
| System.String | message | The message. |
| System.Exception | innerException | The inner exception. |

Implements

System.Runtime.Serialization.ISerializable

System.Runtime.InteropServices._Exception

Class J2kImage

Properties and methods for a JPEG 2000 image.

Inheritance

System.Object
J2kImage

Implements

System.IDisposable

Inherited Members

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

Namespace: [BitMiracle.Jpeg2k](#)
Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public class J2kImage : IDisposable
```

Constructors

J2kImage(Stream)

Initializes a new instance of the [J2kImage](#) class with data from the specified stream.

Declaration

```
public J2kImage(Stream stream)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|------------------|--------|----------------------------|
| System.IO.Stream | stream | The stream with the image. |

Remarks

The stream will not be owned or disposed by this object. The creator of this instance should dispose the stream himself.

J2kImage(Stream, J2kOpenOptions)

Initializes a new instance of the [J2kImage](#) class with data from the specified file.

Declaration

```
public J2kImage(Stream stream, J2kOpenOptions options)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|------|------|-------------|
| | | |

| TYPE | NAME | DESCRIPTION |
|--------------------------------|---------|---|
| System.IO.Stream | stream | The stream with the image. |
| J2kOpenOptions | options | The options to use while opening the image. |

Remarks

The stream will not be owned or disposed by this object. The creator of this instance should dispose the stream himself.

J2kImage(String)

Initializes a new instance of the [J2kImage](#) class with data from the specified file.

Declaration

```
public J2kImage(string fileName)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|---------------|----------|--------------------------------------|
| System.String | fileName | The name of the file with the image. |

J2kImage(String, J2kOpenOptions)

Initializes a new instance of the [J2kImage](#) class with data from the specified stream.

Declaration

```
public J2kImage(string fileName, J2kOpenOptions options)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|--------------------------------|----------|---|
| System.String | fileName | The name of the file with the image. |
| J2kOpenOptions | options | The options to use while opening the image. |

Properties

ColorSpace

Gets color space of the image.

Declaration

```
public J2kColorSpace ColorSpace { get; }
```

Property Value

| TYPE | DESCRIPTION |
|-------------------------------|-------------|
| J2kColorSpace | |

ComponentsInfo

Gets read-only collection of this image components information.

Declaration

```
public ReadOnlyCollection<J2kImageComponentInfo> ComponentsInfo { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--|-------------|
| System.Collections.ObjectModel.ReadOnlyCollection< J2kImageComponentInfo > | |

DefaultTileInfo

Gets information common to all tiles in this image.

Declaration

```
public J2kTileInfo DefaultTileInfo { get; }
```

Property Value

| TYPE | DESCRIPTION |
|-----------------------------|-------------|
| J2kTileInfo | |

Height

Gets height of the image.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

TileCount

Gets number of tiles in this image.

Declaration

```
public int TileCount { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

TileHeight

Gets height of one reference tile.

Declaration

```
public int TileHeight { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

TileWidth

Gets width of one reference tile.

Declaration

```
public int TileWidth { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Width

Gets width of the image.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Methods

Decode()

Decodes all tiles of the image using default decoding options.

Declaration

```
public J2kImageData Decode()
```

Returns

| TYPE | DESCRIPTION |
|------------------------------|-------------------------------|
| J2kImageData | Decoded tiles and properties. |

Decode(J2kDecodingOptions)

Decodes all tiles of the image using specified decoding options.

Declaration

```
public J2kImageData Decode(J2kDecodingOptions options)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|--------------------|---------|--|
| J2kDecodingOptions | options | The options to use while decoding the image tiles. |

Returns

| TYPE | DESCRIPTION |
|--------------|-------------------------------|
| J2kImageData | Decoded tiles and properties. |

DecodeArea(Int32, Int32, Int32, Int32, J2kDecodingOptions)

Decodes part of the image using specified decoding options.

Declaration

```
public J2kImageData DecodeArea(int left, int top, int width, int height, J2kDecodingOptions options)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|--------------------|---------|---|
| System.Int32 | left | The horizontal offset from the origin of the image to the left side of the decoding area. |
| System.Int32 | top | The vertical offset from the origin of the image to the top side of the decoding area. |
| System.Int32 | width | The width of the decoding area. |
| System.Int32 | height | The height of the decoding area. |
| J2kDecodingOptions | options | The options to use while decoding the part of the image. |

Returns

| TYPE | DESCRIPTION |
|--------------|---|
| J2kImageData | Decoded part of the image and properties. |

DecodeTile(Int32)

Decodes one tile of the image using default decoding options.

Declaration

```
public J2kImageData DecodeTile(int tileIndex)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|--------------|-----------|----------------------------------|
| System.Int32 | tileIndex | The index of the tile to decode. |

Returns

| TYPE | DESCRIPTION |
|--------------|-----------------------------------|
| J2kImageData | Decoded tile data and properties. |

DecodeTile(Int32, J2kDecodingOptions)

Decodes one tile of the image using specified decoding options.

Declaration

```
public J2kImageData DecodeTile(int tileIndex, J2kDecodingOptions options)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|--------------------|-----------|---|
| System.Int32 | tileIndex | The index of the tile to decode. |
| J2kDecodingOptions | options | The options to use while decoding the tile. |

Returns

| TYPE | DESCRIPTION |
|--------------|-----------------------------------|
| J2kImageData | Decoded tile data and properties. |

Dispose()

Disposes stream and other resources used by this image.

Declaration

```
public void Dispose()
```

Implements

System.IDisposable

Class J2kImageComponent

Class for a single component of a JPEG 2000 compatible image data.

Inheritance

System.Object
J2kImageComponent

Inherited Members

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

Namespace: [BitMiracle.Jpeg2k](#)
Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public class J2kImageComponent
```

Properties

BitsPerPixel

Gets or sets number of bits per each pixel of this component.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Height

Gets height of this component.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

HorizontalSeparation

Gets the horizontal separation of a sample of the component with respect to the reference grid.

Declaration

```
public int HorizontalSeparation { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Left

Gets horizontal offset from the origin of the whole image to the left side of this component.

Declaration

```
public int Left { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

PixelsAreSigned

Gets a value indicating whether this component pixels are signed or unsigned.

Declaration

```
public bool PixelsAreSigned { get; }
```

Property Value

| TYPE | DESCRIPTION |
|----------------|-------------|
| System.Boolean | |

Top

Gets vertical offset from the origin of the whole image to the top side of this component.

Declaration

```
public int Top { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

VerticalSeparation

Gets the vertical separation of a sample of the component with respect to the reference grid.

Declaration

```
public int VerticalSeparation { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Width

Gets width of this component.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Methods

GetPixels()

Gets access to all component's pixels ordered from left to right, from top to bottom.

Declaration

```
public J2kPixels GetPixels()
```

Returns

| TYPE | DESCRIPTION |
|-----------|-------------|
| J2kPixels | The pixels. |

Class J2kImageComponentInfo

Class for a single component of a JPEG 2000 image.

Inheritance

System.Object
J2kImageComponentInfo

Inherited Members

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

Namespace: [BitMiracle.Jpeg2k](#)
Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public class J2kImageComponentInfo
```

Properties

BitsPerPixel

Gets number of bits per each pixel of this component.

Declaration

```
public int BitsPerPixel { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Height

Gets height of this component.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

HorizontalSeparation

Gets the horizontal separation of a sample of the component with respect to the reference grid.

Declaration

```
public int HorizontalSeparation { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Left

Gets horizontal offset from the origin of the whole image to the left side of this component.

Declaration

```
public int Left { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Top

Gets vertical offset from the origin of the whole image to the top side of this component.

Declaration

```
public int Top { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

VerticalSeparation

Gets the vertical separation of a sample of the component with respect to the reference grid.

Declaration

```
public int VerticalSeparation { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Width

Gets width of this component.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Class J2kImageComponentPrecision

Describes how an image component precision should be changed during decoding.

Inheritance

System.Object
J2kImageComponentPrecision

Inherited Members

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

Namespace: BitMiracle.Jpeg2k
Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public class J2kImageComponentPrecision
```

Properties

Mode

Gets or sets the mode to use while forcing the precision.

Declaration

```
public J2kPrecisionMode Mode { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
|------------------|-------------|
| J2kPrecisionMode | |

Precision

Gets or sets precision (number of bits per each pixel of this component) to force.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Class J2kImageData

Encapsulates information about a JPEG 2000 compatible image data.

Inheritance

System.Object
J2kImageData

Inherited Members

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

Namespace: BitMiracle.Jpeg2k
Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public class J2kImageData
```

Properties

ColorSpace

Gets color space of this image data.

Declaration

```
public J2kColorSpace ColorSpace { get; }
```

Property Value

| TYPE | DESCRIPTION |
|---------------|-------------|
| J2kColorSpace | |

Components

Gets read-only collection of this image data components.

Declaration

```
public ReadOnlyCollection<J2kImageComponent> Components { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--|-------------|
| System.Collections.ObjectModel.ReadOnlyCollection<J2kImageComponent> | |

Height

Gets height of this image data.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Left

Gets horizontal offset from the origin of the reference grid to the left side of this image data.

Declaration

```
public int Left { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Top

Gets vertical offset from the origin of the reference grid to the top side of this image data.

Declaration

```
public int Top { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Width

Gets width of this image data.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Methods

Encode(Stream)

Encodes this image data into JPEG 2000 image using default encoding options and saves the image into the specified stream.

Declaration

```
public void Encode(Stream stream)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|------------------|--------|--|
| System.IO.Stream | stream | The stream to save the JPEG 2000 image to. |

Encode(Stream, J2kEncodingOptions)

Encodes this image data into JPEG 2000 image using specified encoding options and saves the image into the specified stream.

Declaration

```
public void Encode(Stream stream, J2kEncodingOptions options)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|------------------------------------|---------|---|
| System.IO.Stream | stream | The stream to save the JPEG 2000 image to. |
| J2kEncodingOptions | options | The options to use when creating the JPEG 2000 image. |

Encode(String)

Encodes this image data into JPEG 2000 image using default encoding options and saves the image into the specified file.

Declaration

```
public void Encode(string fileName)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|---------------|----------|--|
| System.String | fileName | The name of the file to save the JPEG 2000 image to. |

Encode(String, J2kEncodingOptions)

Encodes this image data into JPEG 2000 image using specified encoding options and saves the image into the specified file.

Declaration

```
public void Encode(string fileName, J2kEncodingOptions options)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|------------------------------------|----------|---|
| System.String | fileName | The name of the file to save the JPEG 2000 image to. |
| J2kEncodingOptions | options | The options to use when creating the JPEG 2000 image. |

FromImage(Stream)

Creates a new instance of the [J2kImageData](#) class by importing image from the specified stream.

Creates a new instance of the [J2kImageData](#) class by importing image from the specified stream.

Declaration

```
public static J2kImageData FromImage(Stream stream)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|------------------|--------|----------------------------|
| System.IO.Stream | stream | The stream with the image. |

Returns

| TYPE | DESCRIPTION |
|------------------------------|---|
| J2kImageData | A new instance of the J2kImageData class with image data from the specified file. |

Remarks

The stream should contain image in one of the supported formats.

At this time supported image formats are BMP and TIFF.

The stream will not be owned or disposed by this object. The caller of this method should dispose the stream himself.

FromImage(Stream, J2kImportOptions)

Creates a new instance of the [J2kImageData](#) class by importing image from the specified stream using specified options.

Declaration

```
public static J2kImageData FromImage(Stream stream, J2kImportOptions options)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|----------------------------------|---------|---|
| System.IO.Stream | stream | The stream with the image. |
| J2kImportOptions | options | The options to use while importing the image. |

Returns

| TYPE | DESCRIPTION |
|------------------------------|---|
| J2kImageData | A new instance of the J2kImageData class with image data from the specified file. |

Remarks

The stream should contain image in one of the supported formats.

At this time supported image formats are BMP and TIFF.

The stream will not be owned or disposed by this object. The caller of this method should dispose the stream himself.

FromImage(String)

Creates a new instance of the [J2kImageData](#) class by importing image from the specified file.

Declaration

```
public static J2kImageData FromImage(string fileName)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|---------------|----------|--------------------------------------|
| System.String | fileName | The name of the file with the image. |

Returns

| TYPE | DESCRIPTION |
|------------------------------|---|
| J2kImageData | A new instance of the J2kImageData class with image data from the specified file. |

Remarks

The file should contain image in one of the supported formats.

At this time supported image formats are BMP and TIFF.

FromImage(String, J2kImportOptions)

Creates a new instance of the [J2kImageData](#) class by importing image from the specified file using specified options.

Declaration

```
public static J2kImageData FromImage(string fileName, J2kImportOptions options)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|----------------------------------|----------|---|
| System.String | fileName | The name of the file with the image. |
| J2kImportOptions | options | The options to use while importing the image. |

Returns

| TYPE | DESCRIPTION |
|------------------------------|---|
| J2kImageData | A new instance of the J2kImageData class with image data from the specified file. |

Remarks

The file should contain image in one of the supported formats.

At this time supported image formats are BMP and TIFF.

Save(Stream, J2kOutputFormat)

Encodes this image data using the specified file format and saves the result to the specified stream.

Declaration

```
public void Save(Stream stream, J2kOutputFormat format)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|------------------|--------|--|
| System.IO.Stream | stream | The stream to save the result to. |
| J2kOutputFormat | format | The file format to use while encoding. |

Save(String, J2kOutputFormat)

Encodes this image data using the specified file format and saves the result to the specified file.

Declaration

```
public void Save(string fileName, J2kOutputFormat format)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|-----------------|----------|---|
| System.String | fileName | The name of the file to save the result to. |
| J2kOutputFormat | format | The file format to use while encoding. |

Class J2kImportOptions

The options to use while creating [J2kImageData](#) from an image in one of the supported image formats.

Inheritance

System.Object
J2kImportOptions

Inherited Members

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

Namespace: [BitMiracle.Jpeg2k](#)
Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public class J2kImportOptions
```

Properties

ImageLeft

Gets or sets image origin offset in x direction.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

ImageTop

Gets or sets image origin offset in y direction.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

SubsamplingDx

Gets or sets subsampling value for dx.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

SubsamplingDy

Gets or sets subsampling value for dy.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Class J2kOpenOptions

The options to be used while opening a JPEG 2000 image.

Inheritance

System.Object
J2kOpenOptions

Inherited Members

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

Namespace: BitMiracle.Jpeg2k
Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public class J2kOpenOptions
```

Properties

IgnoreColorInformation

Gets or sets a value indicating whether color information (palette, channels info etc.) should be ignored.

Declaration

```
public bool IgnoreColorInformation { get; set; }
```

Property Value

| TYPE | DESCRIPTION |
|----------------|-------------|
| System.Boolean | |

QualityLayersToKeep

Gets or sets the number of quality layers to process.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Remarks

If there are less quality layers than the specified number, all the quality layers will be processed.

ResolutionLevelsToDiscard

Gets or sets the number of highest resolution levels to be discarded.

Declaration

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

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Remarks

The image resolution is effectively divided by 2 to the power of the number of discarded levels.

The reduce factor is limited by the smallest total number of decomposition levels among tiles.

If value of this property is not equal to 0, then original dimensions of the image are divided by 2^(value), otherwise (when value is equal to 0) the image is decoded in full resolution.

Enum J2kOutputFormat

Enumeration of all possible output formats

Namespace: [BitMiracle.Jpeg2k](#)

Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public enum J2kOutputFormat
```

Fields

| NAME | DESCRIPTION |
|------|----------------------------------|
| Bmp | Bitmap image file (BMP). |
| Tiff | Tagged Image File Format (TIFF). |

Class J2kPixels

Provides access to pixels in [J2kImageComponent](#).

Inheritance

System.Object
J2kPixels

Inherited Members

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

Namespace: [BitMiracle.Jpeg2k](#)
Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public class J2kPixels
```

Properties

Count

Gets the number of pixels.

Declaration

```
public int Count { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Item[Int32]

Gets the pixel at the specified position.

Declaration

```
public int this[int index] { get; }
```

Parameters

| TYPE | NAME | DESCRIPTION |
|--------------|-------|--|
| System.Int32 | index | Pixel index. Must be greater than or equal to <code>0</code> and less than Count |

Property Value

| TYPE | DESCRIPTION |
|------|-------------|
| | |

| TYPE | DESCRIPTION |
|--------------|--------------------------------------|
| System.Int32 | The pixel at the specified position. |

Enum J2kPrecisionMode

Enumeration of all possible image component precision forcing modes.

Namespace: [BitMiracle.Jpeg2k](#)

Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public enum J2kPrecisionMode
```

Fields

| NAME | DESCRIPTION |
|-------|------------------------------------|
| Clip | Image component should be clipped. |
| Scale | Image component should be scaled. |

Enum J2kProgressionOrder

Enumeration of all possible progression orders.

Namespace: [BitMiracle.Jpeg2k](#)

Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public enum J2kProgressionOrder
```

Fields

| NAME | DESCRIPTION |
|---------|--|
| Cprl | Component-precinct-resolution-layer order |
| Lrcp | Layer-resolution-component-precinct order |
| Pcrl | Precinct-component-resolution-layer order |
| Rlcp | Resolution-layer-component-precinct order |
| Rpcl | Resolution-precinct-component-layer order |
| Unknown | Progression order is not known / not specified |

Enum J2kQualityMode

Supported encoding quality modes.

Namespace: [BitMiracle.Jpeg2k](#)

Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public enum J2kQualityMode
```

Fields

| NAME | DESCRIPTION |
|------------------|---|
| CompressionRatio | Quality values treated as compression ratios for successive layers. |
| FixedQuality | Quality values treated as Peak signal-to-noise ratios (PSNR) for successive layers. |

Enum J2kTileCodingStyle

Enumeration of all possible coding styles.

Namespace: [BitMiracle.Jpeg2k](#)

Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
[Flags]
public enum J2kTileCodingStyle
```

Fields

| NAME | DESCRIPTION |
|----------------------|----------------------------------|
| UseCustomPrecint | Use custom precinct. |
| UseEndOfPacketHeader | Use end of packet header marker. |
| UseStartOfPacket | Use start of packet marker. |

Class J2kTileInfo

Provides information about tile in a JPEG 2000 image.

Inheritance

System.Object
J2kTileInfo

Inherited Members

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

Namespace: BitMiracle.Jpeg2k
Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public class J2kTileInfo
```

Properties

CodingStyle

Gets coding style of this tile.

Declaration

```
public J2kTileCodingStyle CodingStyle { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------------|-------------|
| J2kTileCodingStyle | |

ProgressionOrder

Gets progression order of this tile.

Declaration

```
public J2kProgressionOrder ProgressionOrder { get; }
```

Property Value

| TYPE | DESCRIPTION |
|---------------------|-------------|
| J2kProgressionOrder | |

QualityLayerCount

Gets number of quality layers in this tile.

Declaration

```
public int QualityLayerCount { get; }
```

Property Value

| TYPE | DESCRIPTION |
|--------------|-------------|
| System.Int32 | |

Class LicenseManager

Class for license management. Provides properties and methods to add license to the Jpeg2000.Net library.

Inheritance

System.Object
LicenseManager

Inherited Members

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

Namespace: [BitMiracle.Jpeg2k](#)
Assembly: BitMiracle.Jpeg2000.Net.dll

Syntax

```
public static class LicenseManager
```

Properties

IsValidLicense

Gets a value indicating whether a valid license is applied.

Declaration

```
public static bool IsValidLicense { get; }
```

Property Value

| TYPE | DESCRIPTION |
|----------------|--|
| System.Boolean | <code>true</code> if a valid license is applied; otherwise, <code>false</code> . |

Methods

Reset()

Resets previously set license.

Declaration

```
public static void Reset()
```

SetLicense(String, String)

Applies permanent license.

Declaration

```
public static void SetLicense(string licenseKey, string owner)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|---------------|------------|---------------------------------|
| System.String | licenseKey | License key |
| System.String | owner | Identifier of the license owner |

SetTrialLicense(String)

Applies trial license.

Declaration

```
public static void SetTrialLicense(string licenseKey)
```

Parameters

| TYPE | NAME | DESCRIPTION |
|---------------|------------|-------------|
| System.String | licenseKey | License key |