

JasPer  
2.0.33

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Reference Manual (Version 2.0.33)</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	News . . . . .	1
1.3	License . . . . .	4
1.4	Reporting Bugs . . . . .	5
<b>2</b>	<b>Getting Started</b>	<b>7</b>
2.1	Installation . . . . .	7
2.2	Organization of the Manual . . . . .	9
<b>3</b>	<b>Frequently Asked Questions (FAQ)</b>	<b>11</b>
<b>4</b>	<b>Known Bugs</b>	<b>13</b>
<b>5</b>	<b>Todo List</b>	<b>15</b>
<b>6</b>	<b>Module Index</b>	<b>17</b>
6.1	Modules . . . . .	17
<b>7</b>	<b>Class Index</b>	<b>19</b>
7.1	Class List . . . . .	19
<b>8</b>	<b>File Index</b>	<b>21</b>
8.1	File List . . . . .	21

<b>9</b>	<b>Module Documentation</b>	<b>23</b>
9.1	Initialization and Clean Up	23
9.1.1	Detailed Description	23
9.1.2	Function Documentation	23
9.1.2.1	jas_cleanup()	23
9.1.2.2	jas_init()	24
9.2	I/O Streams	25
9.2.1	Detailed Description	26
9.2.2	Macro Definition Documentation	26
9.2.2.1	jas_stream_clearerr	26
9.2.2.2	jas_stream_eof	27
9.2.2.3	jas_stream_error	27
9.2.2.4	jas_stream_getc	28
9.2.2.5	jas_stream_getrwcoun	28
9.2.2.6	jas_stream_getrwlmit	28
9.2.2.7	jas_stream_peekc	29
9.2.2.8	jas_stream_putc	29
9.2.3	Function Documentation	29
9.2.3.1	jas_stream_close()	29
9.2.3.2	jas_stream_copy()	30
9.2.3.3	jas_stream_display()	30
9.2.3.4	jas_stream_fdopen()	31
9.2.3.5	jas_stream_flush()	31
9.2.3.6	jas_stream_fopen()	32
9.2.3.7	jas_stream_freopen()	32
9.2.3.8	jas_stream_gets()	33
9.2.3.9	jas_stream_gobble()	33
9.2.3.10	jas_stream_isseekable()	34

9.2.3.11	<a href="#">jas_stream_length()</a>	34
9.2.3.12	<a href="#">jas_stream_memopen()</a>	35
9.2.3.13	<a href="#">jas_stream_memopen2()</a>	35
9.2.3.14	<a href="#">jas_stream_pad()</a>	36
9.2.3.15	<a href="#">jas_stream_peek()</a>	36
9.2.3.16	<a href="#">jas_stream_printf()</a>	37
9.2.3.17	<a href="#">jas_stream_puts()</a>	37
9.2.3.18	<a href="#">jas_stream_read()</a>	38
9.2.3.19	<a href="#">jas_stream_rewind()</a>	38
9.2.3.20	<a href="#">jas_stream_seek()</a>	39
9.2.3.21	<a href="#">jas_stream_setrwcounr()</a>	39
9.2.3.22	<a href="#">jas_stream_setrwlirmit()</a>	40
9.2.3.23	<a href="#">jas_stream_tell()</a>	40
9.2.3.24	<a href="#">jas_stream_tmpfile()</a>	41
9.2.3.25	<a href="#">jas_stream_ungetc()</a>	41
9.2.3.26	<a href="#">jas_stream_write()</a>	42
9.3	<a href="#">Image</a>	43
9.3.1	<a href="#">Detailed Description</a>	46
9.3.2	<a href="#">Macro Definition Documentation</a>	46
9.3.2.1	<a href="#">jas_image_brx</a>	46
9.3.2.2	<a href="#">jas_image_bry</a>	46
9.3.2.3	<a href="#">jas_image_clrspc</a>	46
9.3.2.4	<a href="#">jas_image_cmprof</a>	47
9.3.2.5	<a href="#">jas_image_cmptbrx</a>	47
9.3.2.6	<a href="#">jas_image_cmptbry</a>	47
9.3.2.7	<a href="#">jas_image_cmptheight</a>	47
9.3.2.8	<a href="#">jas_image_cmptthstep</a>	48
9.3.2.9	<a href="#">jas_image_cmptprec</a>	48

9.3.2.10	<a href="#">jas_image_cmptsgnd</a>	48
9.3.2.11	<a href="#">jas_image_cmpttlx</a>	48
9.3.2.12	<a href="#">jas_image_cmpttly</a>	48
9.3.2.13	<a href="#">jas_image_cmptvstep</a>	49
9.3.2.14	<a href="#">jas_image_cmptwidth</a>	49
9.3.2.15	<a href="#">jas_image_height</a>	49
9.3.2.16	<a href="#">JAS_IMAGE_MAXFMTS</a>	49
9.3.2.17	<a href="#">jas_image_numcmpts</a>	49
9.3.2.18	<a href="#">jas_image_setclrspc</a>	50
9.3.2.19	<a href="#">jas_image_setcmprof</a>	50
9.3.2.20	<a href="#">jas_image_tlx</a>	50
9.3.2.21	<a href="#">jas_image_tly</a>	50
9.3.2.22	<a href="#">jas_image_width</a>	50
9.3.3	<a href="#">Typedef Documentation</a>	51
9.3.3.1	<a href="#">jas_image_cmpttype_t</a>	51
9.3.3.2	<a href="#">jas_image_colorspc_t</a>	51
9.3.3.3	<a href="#">jas_image_coord_t</a>	51
9.3.3.4	<a href="#">jas_image_smpltype_t</a>	51
9.3.4	<a href="#">Function Documentation</a>	51
9.3.4.1	<a href="#">jas_image_addcmpt()</a>	52
9.3.4.2	<a href="#">jas_image_addfmt()</a>	52
9.3.4.3	<a href="#">jas_image_chclrspc()</a>	52
9.3.4.4	<a href="#">jas_image_clearfmts()</a>	52
9.3.4.5	<a href="#">jas_image_cmpt_domains_same()</a>	53
9.3.4.6	<a href="#">jas_image_copy()</a>	53
9.3.4.7	<a href="#">jas_image_copycmpt()</a>	53
9.3.4.8	<a href="#">jas_image_create()</a>	53
9.3.4.9	<a href="#">jas_image_create0()</a>	53

9.3.4.10	<a href="#">jas_image_decode()</a>	54
9.3.4.11	<a href="#">jas_image_delcmpt()</a>	54
9.3.4.12	<a href="#">jas_image_depalettize()</a>	54
9.3.4.13	<a href="#">jas_image_destroy()</a>	54
9.3.4.14	<a href="#">jas_image_dump()</a>	55
9.3.4.15	<a href="#">jas_image_encode()</a>	55
9.3.4.16	<a href="#">jas_image_fmtfromname()</a>	55
9.3.4.17	<a href="#">jas_image_fmtostr()</a>	55
9.3.4.18	<a href="#">jas_image_getcmptbytype()</a>	55
9.3.4.19	<a href="#">jas_image_getfmt()</a>	56
9.3.4.20	<a href="#">jas_image_ishomosamp()</a>	56
9.3.4.21	<a href="#">jas_image_lookupfmtbyid()</a>	56
9.3.4.22	<a href="#">jas_image_lookupfmtbyname()</a>	56
9.3.4.23	<a href="#">jas_image_rawsize()</a>	56
9.3.4.24	<a href="#">jas_image_readcmpt()</a>	57
9.3.4.25	<a href="#">jas_image_readcmpt2()</a>	57
9.3.4.26	<a href="#">jas_image_readcmptsample()</a>	57
9.3.4.27	<a href="#">jas_image_sampcmpt()</a>	58
9.3.4.28	<a href="#">jas_image_strtofmt()</a>	58
9.3.4.29	<a href="#">jas_image_writecmpt()</a>	58
9.3.4.30	<a href="#">jas_image_writecmpt2()</a>	58
9.3.4.31	<a href="#">jas_image_writecmptsample()</a>	59
<b>10</b>	<b>Class Documentation</b>	<b>61</b>
10.1	<a href="#">jas_image_cmpt_t Struct Reference</a>	61
10.1.1	Detailed Description	61
10.2	<a href="#">jas_image_cmptparm_t Struct Reference</a>	61
10.2.1	Detailed Description	61
10.3	<a href="#">jas_image_fmtinfo_t Struct Reference</a>	62
10.3.1	Detailed Description	62
10.4	<a href="#">jas_image_fmtops_t Struct Reference</a>	62
10.4.1	Detailed Description	62
10.5	<a href="#">jas_image_t Struct Reference</a>	62
10.5.1	Detailed Description	62

<b>11 File Documentation</b>	<b>63</b>
11.1 <a href="#">jas_cm.h File Reference</a>	63
11.1.1 Detailed Description	63
11.2 <a href="#">jas_compiler.h File Reference</a>	63
11.2.1 Detailed Description	63
11.3 <a href="#">jas_debug.h File Reference</a>	63
11.3.1 Detailed Description	64
11.4 <a href="#">jas_dll.h File Reference</a>	64
11.4.1 Detailed Description	64
11.5 <a href="#">jas_fix.h File Reference</a>	64
11.5.1 Detailed Description	64
11.6 <a href="#">jas_getopt.h File Reference</a>	64
11.6.1 Detailed Description	64
11.7 <a href="#">jas_icc.h File Reference</a>	65
11.7.1 Detailed Description	65
11.8 <a href="#">jas_image.h File Reference</a>	65
11.8.1 Detailed Description	68
11.9 <a href="#">jas_init.h File Reference</a>	69
11.9.1 Detailed Description	69
11.10 <a href="#">jas_malloc.h File Reference</a>	69
11.10.1 Detailed Description	69
11.11 <a href="#">jas_math.h File Reference</a>	69
11.11.1 Detailed Description	70
11.12 <a href="#">jas_seq.h File Reference</a>	70
11.12.1 Detailed Description	70
11.13 <a href="#">jas_stream.h File Reference</a>	70
11.13.1 Detailed Description	72
11.14 <a href="#">jas_string.h File Reference</a>	72
11.14.1 Detailed Description	72
11.15 <a href="#">jas_tmr.h File Reference</a>	72
11.15.1 Detailed Description	72
11.16 <a href="#">jas_tvp.h File Reference</a>	72
11.16.1 Detailed Description	73
11.17 <a href="#">jas_types.h File Reference</a>	73
11.17.1 Detailed Description	73
11.18 <a href="#">jas_version.h File Reference</a>	73
11.18.1 Detailed Description	73
11.19 <a href="#">jasper.h File Reference</a>	73
11.19.1 Detailed Description	74



# Chapter 1

## Reference Manual (Version 2.0.33)

### 1.1 Introduction

JasPer is a collection of software (i.e., a library and application programs) for the coding and manipulation of images. This software can handle image data in a variety of formats. One such format supported by JasPer is the JPEG-2000 format defined in ISO/IEC 15444-1. This software was developed by [Michael Adams](#) from the Department of Electrical and Computer Engineering at the University of Victoria, Victoria, BC, Canada.

### 1.2 News

2.0.33 (2021-08-01)  
=====

- \* Fix a JP2/JPC decoder bug.
- \* Fix a build issue impacting some platforms.

2.0.32 (2021-04-18)  
=====

- \* Test release performed with GitHub Actions.

2.0.29 (2021-04-16)  
=====

- \* Loosen some overly tight restrictions on JP2 codestreams, which caused some valid codestreams to be rejected. (#289)

2.0.28 (2021-03-29)  
=====

- \* Fix potential null pointer dereference in the JP2/JPC decoder. (#269)
- \* Fix ignoring of JAS\_STREAM\_FILEOBJ\_NOCLOSE at stream close time. (#286)
- \* Fix integral type sizing problem in JP2 codec. (#284)

2.0.27 (2021-03-18)  
=====

- \* Check for an image containing no samples in the PGX decoder. (#271, #272, #273, #274, #275, #276, #281)
- \* Check for dimensions of zero in the JPC and JPEG decoders.

- \* Fix an arguably incorrect type for an integer literal in the PGX decoder. (#270)
- \* Check for an invalid component reference in the JP2 decoder. (#269)
- \* Check on integer size in JP2 decoder. (#278)

2.0.26 (2021-03-05)  
=====

- \* Fix JP2 decoder bug that can cause a null pointer dereference for some invalid CDEF boxes. (#268) (CVE-2021-3467)

2.0.25 (2021-02-07)  
=====

- \* Fix memory-related bugs in the JPEG-2000 codec resulting from attempting to decode invalid code streams. (#264, #265)  
This fix is associated with CVE-2021-26926 and CVE-2021-26927.
- \* Fix wrong return value under some compilers (#260)
- \* Fix CVE-2021-3272 heap buffer overflow in jp2\_decode (#259)

2.0.24 (2021-01-03)  
=====

- \* Add JAS\_VERSION\_MAJOR, JAS\_VERSION\_MINOR, JAS\_VERSION\_PATCH for easier access to the JasPer version.
- \* Fixes stack overflow bug on Windows, where variable-length arrays are not available. (#256)

2.0.23 (2020-12-08)  
=====

- \* Fix CVE-2020-27828, heap-overflow in cp\_create() in jpc\_enc.c  
<https://github.com/jasper-software/jasper/issues/252>

2.0.22 (2020-10-05)  
=====

- \* Update manual
- \* Remove JPEG dummy codec. Jasper needs libjpeg for JPEG support
- \* Fix test suite build failure regarding disabled MIF codec (#249)
- \* Fix OpenGL/glut detection (#247)

2.0.21 (2020-09-20)  
=====

- \* Fix ZDI-15-529  
<https://github.com/jasper-software/jasper/pull/245>
- \* Fix CVE-2018-19541 in decoder  
<https://github.com/jasper-software/jasper/pull/244>

2.0.20 (2020-09-05)  
=====

- \* Fix several ISO/IEC 15444-4 conformance bugs
- \* Fix new variant of CVE-2016-9398
- \* Disable the MIF codec by default for security reasons (but it is still included in the library);  
in a future release, the MIF codec may also be excluded from the library by default
- \* Add documentation for the I/O streams library API

2.0.19 (2020-07-11)

=====

- \* Fix CVE-2018-9154  
<https://github.com/jasper-software/jasper/issues/215>  
<https://github.com/jasper-software/jasper/issues/166>  
<https://github.com/jasper-software/jasper/issues/175>  
<https://github.com/jasper-maint/jasper/issues/8>
- \* Fix CVE-2018-19541 in encoder  
<https://github.com/jasper-software/jasper/pull/199>  
<https://github.com/jasper-maint/jasper/issues/6>
- \* Fix CVE-2016-9399, CVE-2017-13751  
<https://github.com/jasper-maint/jasper/issues/1>
- \* Fix CVE-2018-19540  
<https://github.com/jasper-software/jasper/issues/182>  
<https://github.com/jasper-maint/jasper/issues/22>
- \* Fix CVE-2018-9055  
<https://github.com/jasper-maint/jasper/issues/9>
- \* Fix CVE-2017-13748  
<https://github.com/jasper-software/jasper/issues/168>
- \* Fix CVE-2017-5503, CVE-2017-5504, CVE-2017-5505  
<https://github.com/jasper-maint/jasper/issues/3>  
<https://github.com/jasper-maint/jasper/issues/4>  
<https://github.com/jasper-maint/jasper/issues/5>  
<https://github.com/jasper-software/jasper/issues/88>  
<https://github.com/jasper-software/jasper/issues/89>  
<https://github.com/jasper-software/jasper/issues/90>
- \* Fix CVE-2018-9252  
<https://github.com/jasper-maint/jasper/issues/16>
- \* Fix CVE-2018-19139  
<https://github.com/jasper-maint/jasper/issues/14>
- \* Fix CVE-2018-19543, CVE-2017-9782  
<https://github.com/jasper-maint/jasper/issues/13>  
<https://github.com/jasper-maint/jasper/issues/18>  
<https://github.com/jasper-software/jasper/issues/140>  
<https://github.com/jasper-software/jasper/issues/182>
- \* Fix CVE-2018-20570  
<https://github.com/jasper-maint/jasper/issues/11>  
<https://github.com/jasper-software/jasper/issues/191>
- \* Fix CVE-2018-20622  
<https://github.com/jasper-maint/jasper/issues/12>  
<https://github.com/jasper-software/jasper/issues/193>
- \* Fix CVE-2016-9398  
<https://github.com/jasper-maint/jasper/issues/10>
- \* Fix CVE-2017-14132  
<https://github.com/jasper-maint/jasper/issues/17>
- \* Fix CVE-2017-5499  
<https://github.com/jasper-maint/jasper/issues/2>  
<https://github.com/jasper-software/jasper/issues/63>
- \* Fix CVE-2018-18873  
<https://github.com/jasper-maint/jasper/issues/15>  
<https://github.com/jasper-software/jasper/issues/184>

- \* Fix <https://github.com/jasper-software/jasper/issues/207>
- \* Fix <https://github.com/jasper-software/jasper/issues/194> part 1
- \* Fix CVE-2017-13750
  - <https://github.com/jasper-software/jasper/issues/165>
  - <https://github.com/jasper-software/jasper/issues/174>
- \* New option `-DJAS_ENABLE_HIDDEN=true` to not export internal symbols in the public symbol table
- \* Fix various memory leaks
- \* Plenty of code cleanups, and performance improvements
- \* Some macros were changed to inline functions. This has the potential to impact some code that made assumptions about the implementation. Some potentially impacted macros include:
  - `jas_matrix_numrows`, `jas_matrix_numcols`
  - `jas_matrix_get`
  - `jas_seq_get`, `jas_seq_start`, `jas_seq_end`
  - `jpc_ms_gettype`
  - `jas_matrix_set` and `jas_seq_set` is affected differently; the old macro was an actual expression returning the value, while the new function does not.
 The following macros have been changed, too, but are likely not affected, since they have been an rvalue-expression anyway:
  - `JP2_DTYPETOBCP`, `JP2_BPCTODTYPE`
  - `JAS_IMAGE_CDT_{SETSGND,GETSGND,SETPREC,GETPREC}`
  - `jas_image_cmptdtype`
  - macros from here
  - `jas_matrix_setv`, `jas_matrix_getvref`
  - `jas_matrix_bind{row,col}`
  - the `jpc_fix` family
  - the `JPC_QCX` and `JPC_COX` families

## 1.3 License

JasPer License Version 2.0

Copyright (c) 2001-2016 Michael David Adams  
 Copyright (c) 1999-2000 Image Power, Inc.  
 Copyright (c) 1999-2000 The University of British Columbia

All rights reserved.

Permission is hereby granted, free of charge, to any person (the "User") obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

1. The above copyright notices and this permission notice (which includes the disclaimer below) shall be included in all copies or substantial portions of the Software.
2. The name of a copyright holder shall not be used to endorse or promote products derived from the Software without specific prior written permission.

THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS

"AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY. EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH, PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.

## 1.4 Reporting Bugs

All bug reports should be submitted via the issue-tracking system provided by GitHub. To submit a bug report, go to the following URL and click on the "New issue" button:

<https://github.com/jasper-software/jasper/issues>

Please do not submit bug reports directly to the author of JasPer via email, as bug reports that are not submitted via the above issue-tracking system on GitHub are easy to be lost.



## Chapter 2

# Getting Started

The following sections are useful for getting started with the Jasper software:

- [Installation](#). Describes how to install the Jasper software.
- [Organization of the Manual](#). Briefly explains the organization of the manual (e.g., where to find things).

## 2.1 Installation

Installation  
=====

The process required to install Jasper is described below.

Installation on Systems Running Unix  
-----

In what follows, let `$SOURCE_DIR` denote the top-level directory of the Jasper software source tree (i.e., the directory containing the files named `LICENSE` and `INSTALL`) and let `$INSTALL_DIR` denote the target directory for installation.

1) Select an empty directory to use for building the software. Let `$BUILD_DIR` denote this directory.

2) Generate the makefiles used for building the software. To do this, invoke the command:

```
cmake -G "Unix Makefiles" -H$SOURCE_DIR -B$BUILD_DIR \
      -DCMAKE_INSTALL_PREFIX=$INSTALL_DIR $OPTIONS
```

where `$OPTIONS` corresponds to zero or more `-D` options as described below under the heading "Cmake Options".

3) Change the working directory to the build directory. To do this, use the command:

```
cd $BUILD_DIR
```

4) Build the code using the make utility. To do this, invoke the command:

```
make clean all
```

5) Run the test suite to ensure that the software seems to be working correctly. To do this, invoke the command:

```
make test
```

If more verbose output from the testing process is desired (e.g., to assist in diagnosing a problem), instead use the command:

```
make test ARGS="-V"
```

6) Install the software. To do this, invoke the command:

```
make install
```

#### Additional Remarks:

When building the Jasper software under Mac OSX, only the use of the native framework for OpenGL is officially supported. If the Freeglut library is installed on your system, you will need to ensure that the native GLUT library (as opposed to the Freeglut library) is used by the build process. This can be accomplished by adding an extra option to the cmake command line that resembles the following:

```
-DGLUT_glut_LIBRARY=/System/Library/Frameworks/GLUT.framework
```

#### Installation on Systems Running Microsoft Windows

In what follows, let %SOURCE\_DIR% denote the top-level directory of the Jasper software source tree (i.e., the directory containing the files named LICENSE and INSTALL) and let %INSTALL\_DIR% denote the target directory for installation.

1) Select an empty directory to use for building the software. Let %BUILD\_DIR% denote this directory.

2) Generate the project file needed to build the software with Microsoft Visual Studio. To do this, invoke the command:

```
cmake -G "Visual Studio 12 2013 Win64" -H%SOURCE_DIR% -B%BUILD_DIR% ^
-DCMAKE_INSTALL_PREFIX=%INSTALL_DIR% %OPTIONS%
```

where %OPTIONS% corresponds to zero or more -D options as described below under the heading "Cmake Options". (Note the caret symbol "^" above denotes line continuation.)

3) Build and install the software. To do this, invoke the command:

```
msbuild %build_dir%\INSTALL.vcxproj
```

#### Cmake Options

The option OPTION can be set to the value VALUE with a command-line option of the form -DOPTION=VALUE  
The following options are supported:

CMAKE\_INSTALL\_PREFIX  
Specify the installation directory.  
Value: A directory name.

CMAKE\_BUILD\_TYPE  
Specify the build type (i.e., release or debug).  
Valid values: Debug or Release

JAS\_ENABLE\_DOC



Enable the building of the documentation (which requires LaTeX).  
Valid values: true and false

JAS\_ENABLE\_LIBJPEG  
Enable the use of the JPEG library  
Valid values: true and false

JAS\_ENABLE\_OPENGL  
Enable the use of the OpenGL and GLUT libraries.  
Valid values: true and false

JAS\_ENABLE\_SHARED  
Enable the building of shared libraries.  
Valid values: true or false

JAS\_ENABLE\_HIDDEN  
Hide internal symbols? Enabling this results in a smaller binary.  
Valid values: true or false

JAS\_ENABLE\_32BIT  
Force the use of 32 bit integers? On 64 bit CPUs, JasPer historically used 64 bit integers which consumes more memory, is slower and has no advantages. This produces a different ABI, so the resulting library is not compatible with other builds.  
Valid values: true or false

JAS\_ENABLE\_ASAN  
Enable the Address Sanitizer.  
Valid values: true or false

JAS\_ENABLE\_USAN  
Enable the Undefined-Behavior Sanitizer.  
Valid values: true or false

JAS\_ENABLE\_LSAN  
Enable the Leak Sanitizer.  
Valid values: true or false

JAS\_ENABLE\_MSAN  
Enable the Memory Sanitizer.  
Valid values: true or false

## 2.2 Organization of the Manual

The library is partitioned into groups of related code called modules. The documentation is also partitioned in this way. The documentation for each of the various modules can be found in the [modules page](#).



## Chapter 3

# Frequently Asked Questions (FAQ)

The following is a list of common questions/problems encountered when using the library.

-



## Chapter 4

# Known Bugs

All bugs reported in Jasper are tracked using the issue-tracking functionality provided by GitHub. If you encounter a problem with Jasper and you would like to know if it is a known problem, please check the issue tracker for Jasper on GitHub, which can be found at the following URL:

`https://github.com/jasper-software/jasper/issues`

If you happen to find a bug that has not been previously reported, please report it so that it can be fixed. New bugs can be reported by creating a new issue using the page at the above URL.



## Chapter 5

## Todo List

**Member `jas_stream_clearerr` (`stream`)**

TODO/FIXME: Should this macro evaluate to void?

**Member `jas_stream_copy` (`jas_stream_t *destination`, `jas_stream_t *source`, `int count`)**

TODO/FIXME: count should probably be a `size_t`; return type `ssize_t`?

**Member `jas_stream_display` (`jas_stream_t *stream`, `FILE *fp`, `int count`)**

TODO/FIXME: should count be unsigned int or `size_t` instead of int?

**Member `jas_stream_pad` (`jas_stream_t *stream`, `int count`, `int value`)**

TODO: should the count be `size_t`; return type maybe `size_t`?

**Member `jas_stream_printf` (`jas_stream_t *stream`, `const char *format`,...)**

I think that the return type of int is okay here. It is consistent with printf and friends.

**Member `jas_stream_setrwcoun` (`jas_stream_t *stream`, `long rw_count`)**

TODO/FIXME: Should this macro evaluate to void?





## Chapter 6

# Module Index

### 6.1 Modules

Here is a list of all modules:

Initialization and Clean Up . . . . .	<a href="#">23</a>
I/O Streams . . . . .	<a href="#">25</a>
Image . . . . .	<a href="#">43</a>



## Chapter 7

# Class Index

### 7.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">jas_image_cmpt_t</a>	Image component class . . . . .	61
<a href="#">jas_image_cmptparm_t</a>	Component parameters class . . . . .	61
<a href="#">jas_image_fmtnfo_t</a>	Image format information . . . . .	62
<a href="#">jas_image_fmtnops_t</a>	Image format-dependent operations . . . . .	62
<a href="#">jas_image_t</a>	Image class . . . . .	62



## Chapter 8

# File Index

### 8.1 File List

Here is a list of all documented files with brief descriptions:

<b>bmp_cod.h</b>	??
<b>bmp_enc.h</b>	??
<a href="#">jas_cm.h</a>	
JasPer Color Management	63
<a href="#">jas_compiler.h</a>	
Compiler-related macros	63
<a href="#">jas_debug.h</a>	
JasPer Debugging-Related Functionality	63
<a href="#">jas_dll.h</a>	
Shared Library Macros	64
<a href="#">jas_fix.h</a>	
JasPer Fixed-Point Number Class	64
<a href="#">jas_getopt.h</a>	
Command Line Option Parsing Code	64
<a href="#">jas_icc.h</a>	
ICC Profile	65
<a href="#">jas_image.h</a>	
JasPer Image Class	65
<a href="#">jas_init.h</a>	
JasPer Initialization/Cleanup Code	69
<a href="#">jas_malloc.h</a>	
JasPer Memory Allocator	69
<a href="#">jas_math.h</a>	
Math-Related Code	69
<a href="#">jas_seq.h</a>	
Sequence/Matrix Library	70
<a href="#">jas_stream.h</a>	
I/O Stream Class	70
<a href="#">jas_string.h</a>	
String Library	72
<a href="#">jas_tmr.h</a>	
Timer Code	72

<a href="#">jas_tvp.h</a>	Tag/Value Pair Parser . . . . .	72
<a href="#">jas_types.h</a>	Primitive Types . . . . .	73
<a href="#">jas_version.h</a>	JasPer Version . . . . .	73
<a href="#">jasper.h</a>	JasPer Main Header . . . . .	73
<a href="#">jp2_cod.h</a>		??
<a href="#">jp2_dec.h</a>		??
<a href="#">jpc_bs.h</a>		??
<a href="#">jpc_cod.h</a>		??
<a href="#">jpc_cs.h</a>		??
<a href="#">jpc_dec.h</a>		??
<a href="#">jpc_enc.h</a>		??
<a href="#">jpc_fix.h</a>		??
<a href="#">jpcflt.h</a>		??
<a href="#">jpc_math.h</a>		??
<a href="#">jpc_mct.h</a>		??
<a href="#">jpc_mqcod.h</a>		??
<a href="#">jpc_mqdec.h</a>		??
<a href="#">jpc_mqenc.h</a>		??
<a href="#">jpc_qmfb.h</a>		??
<a href="#">jpc_t1cod.h</a>		??
<a href="#">jpc_t1dec.h</a>		??
<a href="#">jpc_t1enc.h</a>		??
<a href="#">jpc_t2cod.h</a>		??
<a href="#">jpc_t2dec.h</a>		??
<a href="#">jpc_t2enc.h</a>		??
<a href="#">jpc_tagtree.h</a>		??
<a href="#">jpc_tsfb.h</a>		??
<a href="#">jpc_util.h</a>		??
<a href="#">jpg_cod.h</a>		??
<a href="#">jpg_enc.h</a>		??
<a href="#">jpg_jpeglib.h</a>		??
<a href="#">mif_cod.h</a>		??
<a href="#">pgx_cod.h</a>		??
<a href="#">pgx_enc.h</a>		??
<a href="#">pnm_cod.h</a>		??
<a href="#">pnm_enc.h</a>		??
<a href="#">ras_cod.h</a>		??
<a href="#">ras_enc.h</a>		??

## Chapter 9

# Module Documentation

### 9.1 Initialization and Clean Up

Initialization and Clean Up.

#### Functions

- JAS\_DLLEXPORT int [jas\\_init](#) (void)  
*Initialize the JasPer library.*
- JAS\_DLLEXPORT void [jas\\_cleanup](#) (void)  
*Perform any clean up for the JasPer library.*

#### 9.1.1 Detailed Description

Initialization and Clean Up.

#### 9.1.2 Function Documentation

##### 9.1.2.1 [jas\\_cleanup\(\)](#)

```
JAS_DLLEXPORT void jas_cleanup (  
    void )
```

Perform any clean up for the JasPer library.

This function performs any clean up for the JasPer library.

### 9.1.2.2 jas\_init()

```
JAS_DLLEXPORT int jas_init (  
    void )
```

Initialize the JasPer library.

This function must be called before any other code in the JasPer library is invoked. This function registers the codecs that are enabled by default.

#### Returns

If successful, zero is returned; otherwise, a nonzero value is returned.



## 9.2 I/O Streams

I/O streams.

### Macros

- `#define jas_stream_eof(stream) (((stream)->flags_ & JAS_STREAM_EOF) != 0)`  
*Get the EOF indicator for a stream.*
- `#define jas_stream_error(stream) (((stream)->flags_ & JAS_STREAM_ERR) != 0)`  
*Get the error indicator for a stream.*
- `#define jas_stream_clearerr(stream) ((stream)->flags_ &= ~(JAS_STREAM_ERR | JAS_STREAM_EOF))`  
*Clear the error indicator for a stream.*
- `#define jas_stream_getrwlimit(stream) (((const jas_stream_t*)(stream))->rwlimit_)`  
*Get the read/write limit for a stream.*
- `#define jas_stream_getrwcoun(stream) (((const jas_stream_t*)(stream))->rwcoun_)`  
*Get the read/write count for a stream.*
- `#define jas_stream_getc(stream) jas_stream_getc_func(stream)`  
*jas\_stream\_getc Read a character from a stream.*
- `#define jas_stream_putc(stream, c) jas_stream_putc_func(stream, c)`  
*jas\_stream\_putc Write a character to a stream.*
- `#define jas_stream_peekc(stream)`  
*Look at the next character to be read from a stream without actually removing the character from the stream.*

### Functions

- `JAS_DLLEXPORT jas_stream_t * jas_stream_fopen (const char *filename, const char *mode)`  
*Open a file as a stream.*
- `JAS_DLLEXPORT jas_stream_t * jas_stream_memopen (char *buffer, int buffer_size)`  
*Open a memory buffer as a stream.*
- `JAS_DLLEXPORT jas_stream_t * jas_stream_memopen2 (char *buffer, size_t buffer_size)`
- `JAS_DLLEXPORT jas_stream_t * jas_stream_fdopen (int fd, const char *mode)`  
*Open a file descriptor as a stream.*
- `JAS_DLLEXPORT jas_stream_t * jas_stream_freopen (const char *path, const char *mode, FILE *fp)`  
*Open a stdio (i.e., C standard library) stream as a stream.*
- `JAS_DLLEXPORT jas_stream_t * jas_stream_tmpfile (void)`  
*Open a temporary file as a stream.*
- `JAS_DLLEXPORT int jas_stream_close (jas_stream_t *stream)`  
*Close a stream.*
- `JAS_DLLEXPORT long jas_stream_setrwlimit (jas_stream_t *stream, long rwlimit)`  
*Set the read/write limit for a stream.*
- `JAS_DLLEXPORT long jas_stream_setrwcoun (jas_stream_t *stream, long rw_coun)`  
*Set the read/write count for a stream.*
- `JAS_DLLEXPORT unsigned jas_stream_read (jas_stream_t *stream, void *buffer, unsigned count)`  
*Read characters from a stream into a buffer.*
- `JAS_DLLEXPORT unsigned jas_stream_peek (jas_stream_t *stream, void *buffer, size_t count)`

Attempt to retrieve one or more pending characters of input from a stream into a buffer without actually removing the characters from the stream.

- JAS\_DLLEXPORT unsigned [jas\\_stream\\_write](#) (jas\_stream\_t \*stream, const void \*buffer, unsigned count)

Write characters from a buffer to a stream.

- JAS\_DLLEXPORT int [jas\\_stream\\_printf](#) (jas\_stream\_t \*stream, const char \*format,...)

Write formatted output to a stream.

- JAS\_DLLEXPORT int [jas\\_stream\\_puts](#) (jas\_stream\_t \*stream, const char \*s)

Write a string to a stream.

- JAS\_DLLEXPORT char \* [jas\\_stream\\_gets](#) (jas\_stream\_t \*stream, char \*buffer, int buffer\_size)

Read a line of input from a stream.

- JAS\_DLLEXPORT int [jas\\_stream\\_ungetc](#) (jas\_stream\_t \*stream, int c)

Put a character back on a stream.

- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT int [jas\\_stream\\_isseekable](#) (jas\_stream\_t \*stream)

Determine if stream supports seeking.

- JAS\_DLLEXPORT long [jas\\_stream\\_seek](#) (jas\_stream\_t \*stream, long offset, int origin)

Set the current position within the stream.

- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT long [jas\\_stream\\_tell](#) (jas\_stream\_t \*stream)

Get the current position within the stream.

- JAS\_DLLEXPORT int [jas\\_stream\\_rewind](#) (jas\_stream\_t \*stream)

Seek to the beginning of a stream.

- JAS\_DLLEXPORT int [jas\\_stream\\_flush](#) (jas\_stream\_t \*stream)

Flush any pending output to a stream.

- JAS\_DLLEXPORT int [jas\\_stream\\_copy](#) (jas\_stream\_t \*destination, jas\_stream\_t \*source, int count)

Copy data from one stream to another.

- JAS\_DLLEXPORT int [jas\\_stream\\_display](#) (jas\_stream\_t \*stream, FILE \*fp, int count)

Print a hex dump of data read from a stream.

- JAS\_DLLEXPORT int [jas\\_stream\\_gobble](#) (jas\_stream\_t \*stream, int count)

Consume (i.e., discard) characters from stream.

- JAS\_DLLEXPORT int [jas\\_stream\\_pad](#) (jas\_stream\_t \*stream, int count, int value)

Write a fill character multiple times to a stream.

- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT long [jas\\_stream\\_length](#) (jas\_stream\_t \*stream)

Get the size of the file associated with the specified stream.

## 9.2.1 Detailed Description

I/O streams.

For more detail on I/O streams, please refer to [here](#).

## 9.2.2 Macro Definition Documentation

### 9.2.2.1 [jas\\_stream\\_clearerr](#)

```
#define jas_stream_clearerr(  
    stream ) ((stream)->flags_ &= ~(JAS_STREAM_ERR | JAS_STREAM_EOF))
```

Clear the error indicator for a stream.

**Parameters**

<i>stream</i>	The stream whose error indicator is to be cleared.
---------------	--

**Todo** TODO/FIXME: Should this macro evaluate to void?

**9.2.2.2 jas\_stream\_eof**

```
#define jas_stream_eof(  
    stream ) (((stream)->flags_ & JAS_STREAM_EOF) != 0)
```

Get the EOF indicator for a stream.

**Parameters**

<i>stream</i>	The stream whose EOF indicator is to be queried.
---------------	--

**Returns**

The value of the EOF indicator is returned. A nonzero value indicates that the stream has encountered EOF.

**9.2.2.3 jas\_stream\_error**

```
#define jas_stream_error(  
    stream ) (((stream)->flags_ & JAS_STREAM_ERR) != 0)
```

Get the error indicator for a stream.

**Parameters**

<i>stream</i>	The stream whose error indicator is to be queried.
---------------	--

**Returns**

The value of the error indicator is returned. A nonzero value indicates that the stream has encountered an error of some type (such as an I/O error). Note that EOF is not an error.

#### 9.2.2.4 `jas_stream_getc`

```
#define jas_stream_getc(  
    stream ) jas_stream_getc_func(stream)
```

`jas_stream_getc` Read a character from a stream.

#### 9.2.2.5 `jas_stream_getrwc`

```
#define jas_stream_getrwc(  
    stream ) (((const jas_stream_t *) (stream))->rwcnt_)
```

Get the read/write count for a stream.

##### Parameters

<i>stream</i>	A pointer to the stream whose read/write count is to be queried.
---------------	--

##### Returns

The read/write count is returned. This operation cannot fail.

#### 9.2.2.6 `jas_stream_getrwl`

```
#define jas_stream_getrwl(  
    stream ) (((const jas_stream_t *) (stream))->rwlimit_)
```

Get the read/write limit for a stream.

##### Parameters

<i>stream</i>	A pointer to the stream whose read/write limit is to be queried.
---------------	--

##### Returns

The read/write limit for the stream is returned. This operation cannot fail. A negative read/write limit indicates no limit (i.e., an limit that is effectively infinite).

### 9.2.2.7 `jas_stream_peekc`

```
#define jas_stream_peekc(  
    stream )
```

**Value:**

```
((stream)->cnt_ <= 0) ? jas_stream_fillbuf(stream, 0) : \  
    ((int) (* (stream)->ptr_))
```

Look at the next character to be read from a stream without actually removing the character from the stream.

**Parameters**

<i>stream</i>	A pointer to the stream to be examined.
---------------	---

This function examines the next character that would be read from the stream and returns this character without actually removing it from the stream.

**Returns**

If the peek operation fails (e.g., due to EOF or I/O error), EOF is returned. Otherwise, the character that would be next read from the stream is returned.

### 9.2.2.8 `jas_stream_putc`

```
#define jas_stream_putc(  
    stream,  
    c ) jas_stream_putc_func(stream, c)
```

`jas_stream_putc` Write a character to a stream.

## 9.2.3 Function Documentation

### 9.2.3.1 `jas_stream_close()`

```
JAS_DLLEXPORT int jas_stream_close (  
    jas_stream_t * stream )
```

Close a stream.

**Parameters**

<i>stream</i>	A (nonnull) pointer to the stream to be closed.
---------------	---

The close operation will implicitly flush any pending output to the stream before closing. If such a flush operation fails, this will be reflected in the return value of this function. For many systems, it is likely that the main reason that this function can fail is due to an I/O error when flushing buffered output.

**Returns**

If no errors are encountered when closing the stream, 0 is returned. Otherwise, a nonzero value is returned.

**9.2.3.2 `jas_stream_copy()`**

```
JAS_DLLEXPORT int jas_stream_copy (
    jas_stream_t * destination,
    jas_stream_t * source,
    int count )
```

Copy data from one stream to another.

**Parameters**

<i>destination</i>	A pointer to the stream that is the destination for the copy.
<i>source</i>	A pointer to the stream that is the source for the copy.
<i>count</i>	The number of characters to copy.

The function copies the specified number of characters from the source stream to the destination stream.

**Returns**

Upon success, 0 is returned; otherwise, -1 is returned.

**Todo** TODO/FIXME: count should probably be a `size_t`; return type `ssize_t`?

**9.2.3.3 `jas_stream_display()`**

```
JAS_DLLEXPORT int jas_stream_display (
    jas_stream_t * stream,
    FILE * fp,
    int count )
```

Print a hex dump of data read from a stream.

**Parameters**

<i>stream</i>	A pointer to the stream from which to read data.
<i>fp</i>	A pointer to a stdio stream (i.e., FILE) to which to print the hex dump.
<i>count</i>	The number of characters to include in the hex dump.

This function prints a hex dump of data read from a stream to a stdio stream. This function is most likely to be useful for debugging.

**Returns**

Upon success, 0 is returned. Otherwise, a negative value is returned.

**Todo** TODO/FIXME: should count be unsigned int or size\_t instead of int?

**9.2.3.4 jas\_stream\_fdopen()**

```
JAS_DLLEXPORT jas_stream_t* jas_stream_fdopen (
    int fd,
    const char * mode )
```

Open a file descriptor as a stream.

**Parameters**

<i>fd</i>	The file descriptor of the file to open as a stream.
<i>mode</i>	A pointer to a string specifying the open mode. The format of this string is similar to that of the fdopen function in the C standard library.

**Returns**

Upon success, a pointer to the opened stream is returned. Otherwise, a null pointer is returned.

**9.2.3.5 jas\_stream\_flush()**

```
JAS_DLLEXPORT int jas_stream_flush (
    jas_stream_t * stream )
```

Flush any pending output to a stream.

**Parameters**

<i>stream</i>	A pointer to the stream for which output should be flushed.
---------------	---

The function flushes any buffered output to the underlying file object.

**Returns**

Upon success, zero is returned. Otherwise, a negative value is returned.

**9.2.3.6 jas\_stream\_fopen()**

```
JAS_DLLEXPORT jas_stream_t* jas_stream_fopen (
    const char * filename,
    const char * mode )
```

Open a file as a stream.

**Parameters**

<i>filename</i>	A pointer to the pathname of the file to be opened.
<i>mode</i>	A pointer to the string specifying the open mode. The open mode is similar to that used by the fopen function in the C standard library.

**Returns**

Upon success, a pointer to the opened stream is returned. Otherwise, a null pointer is returned.

**9.2.3.7 jas\_stream\_freopen()**

```
JAS_DLLEXPORT jas_stream_t* jas_stream_freopen (
    const char * path,
    const char * mode,
    FILE * fp )
```

Open a stdio (i.e., C standard library) stream as a stream.

**Parameters**

<i>path</i>	A pointer to a string containing the path of the filename associated with the stdio stream.
<i>mode</i>	A pointer to a string containing the open mode to be used for the (JasPer) stream. This string is similar to that used by the fdopen function in the C standard library.
<i>fp</i>	A pointer to the stdio stream.



It is unspecified whether the open mode specified by mode can be changed from the open mode used for opening the stdio stream.

#### Returns

Upon success, a pointer to the opened stream is returned. Otherwise, a null pointer is returned.

#### 9.2.3.8 `jas_stream_gets()`

```
JAS_DLLEXPORT char* jas_stream_gets (
    jas_stream_t * stream,
    char * buffer,
    int buffer_size )
```

Read a line of input from a stream.

##### Parameters

<i>stream</i>	A pointer to the stream from which to read input.
<i>buffer</i>	A pointer to the start of the buffer to hold to input to be read.
<i>buffer_size</i>	The size of the buffer in characters.

The function reads a line of input from a stream into a buffer. If a newline character is read, it is placed in the buffer. Since the buffer may be too small to hold the input, this operation can fail due to attempted buffer overrun.

#### Returns

If the operation fails (e.g., due to an I/O error or attempted buffer overrun), a null pointer is returned. Otherwise, buffer is returned.

#### 9.2.3.9 `jas_stream_gobble()`

```
JAS_DLLEXPORT int jas_stream_gobble (
    jas_stream_t * stream,
    int count )
```

Consume (i.e., discard) characters from stream.

##### Parameters

<i>stream</i>	A pointer to the stream from which to discard data.
<i>count</i>	The number of characters to discard.

This function reads and discards the specified number of characters from the given stream.

#### Returns

This function returns the number of characters read and discarded. If an error or EOF is encountered, the number of characters read will be less than count. To distinguish EOF from an I/O error, [jas\\_stream\\_eof\(\)](#) and [jas\\_stream\\_error\(\)](#) can be used.

#### Warning

TODO/FIXME: count be size\_t and return type should be ssize\_t

#### 9.2.3.10 [jas\\_stream\\_isseekable\(\)](#)

```
JAS_ATTRIBUTE_PURE JAS_DLLEXPORT int jas_stream_isseekable (
    jas_stream_t * stream )
```

Determine if stream supports seeking.

#### Parameters

<i>stream</i>	A pointer to the stream to query.
---------------	-----------------------------------

The function is a predicate that tests if the underlying file object supports seek operations.

#### Returns

If the underlying file object supports seek operations, a (strictly) positive value is returned. Otherwise, 0 is returned.

#### 9.2.3.11 [jas\\_stream\\_length\(\)](#)

```
JAS_ATTRIBUTE_PURE JAS_DLLEXPORT long jas_stream_length (
    jas_stream_t * stream )
```

Get the size of the file associated with the specified stream.

#### Parameters

<i>stream</i>	
---------------	--

This function queries the size (i.e., length) of the underlying file object associated with the specified stream. The specified stream must be seekable.

**Returns**

Upon success, the size of the stream is returned. If an error occurs, a negative value is returned.

**Warning**

TODO/FIXME: the return type should be `ssize_t`?

**9.2.3.12 `jas_stream_memopen()`**

```
JAS_DLLEXPORT jas_stream_t* jas_stream_memopen (
    char * buffer,
    int buffer_size )
```

Open a memory buffer as a stream.

**Parameters**

<i>buffer</i>	A pointer to the buffer to be used to store stream data.
<i>buffer_size</i>	The size of the buffer.

- If `buffer` is 0 and `buffer_size` > 0: a buffer is dynamically allocated with size `buffer_size` and this buffer is not growable.
- If `buffer` is 0 and `buffer_size` is 0: a buffer is dynamically allocated whose size will automatically grow to accommodate the amount of data written.
- If `buffer` is not 0: `buffer_size` (which, in this case, is not currently allowed to be zero) is the size of the (nongrowable) buffer pointed to by `buffer`.

**Warning**

TODO/FIXME: The type of the `buffer_size` parameter will be changed to `size_t` in the future.

TODO/FIXME: In a later release, this function will be changed to have the same prototype as `jas_stream_memopen2`, at which point `jas_stream_memopen` will be removed.

**9.2.3.13 `jas_stream_memopen2()`**

```
JAS_DLLEXPORT jas_stream_t* jas_stream_memopen2 (
    char * buffer,
    size_t buffer_size )
```

**Warning**

This function will be renamed `jas_stream_memopen` in a future release. Do not use this function.

#### 9.2.3.14 `jas_stream_pad()`

```
JAS_DLLEXPORT int jas_stream_pad (
    jas_stream_t * stream,
    int count,
    int value )
```

Write a fill character multiple times to a stream.

##### Parameters

<i>stream</i>	A pointer to the stream to which to write.
<i>count</i>	The number of times to write the fill character to the stream.
<i>value</i>	The fill character.

This function writes the given fill character to a stream a specified number of times. If a count of zero is specified, the function should have no effect.

##### Returns

The number of times the fill character was written to the stream is returned. If this value is less than the specified count, an error must have occurred.

**Todo** TODO: should the count be `size_t`; return type maybe `size_t`?

#### 9.2.3.15 `jas_stream_peek()`

```
JAS_DLLEXPORT unsigned jas_stream_peek (
    jas_stream_t * stream,
    void * buffer,
    size_t count )
```

Attempt to retrieve one or more pending characters of input from a stream into a buffer without actually removing the characters from the stream.

##### Parameters

<i>stream</i>	A pointer to the stream from which to retrieve pending input.
<i>buffer</i>	A pointer to the start of the buffer.
<i>count</i>	A count of how many characters to retrieve.

The extent to which one can peek into the stream is limited. Therefore, this function can fail if count is sufficiently large.

**Returns**

Returns the number of bytes copied to the given buffer, or 0 on error or EOF.

**Warning**

TODO/FIXME: peeking at EOF should be distinguishable from an I/O error

**9.2.3.16 jas\_stream\_printf()**

```
JAS_DLLEXPORT int jas_stream_printf (
    jas_stream_t * stream,
    const char * format,
    ... )
```

Write formatted output to a stream.

**Parameters**

<i>stream</i>	A pointer to the stream to which to write output.
<i>format</i>	A pointer to a format string similar to the printf function in the C standard library.

The function prints the information associated with the format string to the specified stream.

**Returns**

Upon success, the number of characters output to the stream is returned. If an error is encountered, a negative value is returned.

**Todo** I think that the return type of int is okay here. It is consistent with printf and friends.

**9.2.3.17 jas\_stream\_puts()**

```
JAS_DLLEXPORT int jas_stream_puts (
    jas_stream_t * stream,
    const char * s )
```

Write a string to a stream.

**Parameters**

<i>stream</i>	A pointer to the stream to which the string should be written.
<i>s</i>	A pointer to a null-terminated string for output.

The null character is not output.

#### Returns

Upon success, a nonnegative value is returned. Upon failure, a negative value is returned.

#### 9.2.3.18 `jas_stream_read()`

```
JAS_DLLEXPORT unsigned jas_stream_read (
    jas_stream_t * stream,
    void * buffer,
    unsigned count )
```

Read characters from a stream into a buffer.

#### Parameters

<i>stream</i>	A pointer to the stream from which to read data.
<i>buffer</i>	A pointer to the start of the buffer.
<i>count</i>	A count of the number of characters to read (nominally).

If `count` is zero, the function has no effect (and therefore cannot fail). Otherwise, the function attempts to read `count` characters from the stream `stream` into the buffer starting at `buffer`. The number of characters read can be less than `count`, due to end-of-file (EOF) or an I/O error.

#### Returns

The number of characters read is returned. In the case that the number of characters read is less than `count`, [`jas\_stream\_eof\(\)`](#) and/or [`jas\_stream\_error\(\)`](#) must be used to distinguish between:

1. a failure due to an I/O error
2. a failure due to the read/write limit being exceeded
3. EOF.

TODO/CHECK: can items 1 and 2 be distinguished currently?

#### Warning

TODO/FIXME/CHECK: `jas_stream_error` should be true if RWLIMIT exceeded? or need a `jas_stream_rwlimit` predicate?

TODO/FIXME: In the future, the type of the count parameter and the return type will be changed to `size_t`.

#### 9.2.3.19 `jas_stream_rewind()`

```
JAS_DLLEXPORT int jas_stream_rewind (
    jas_stream_t * stream )
```

Seek to the beginning of a stream.

**Parameters**

<i>stream</i>	A pointer to the stream whose position is to be set.
---------------	--

The stream position is set to the start of the stream. This function is equivalent to returning the value of `jas_stream_↵seek(stream, 0, SEEK_SET)`.

**Returns**

Upon success, the new stream position is returned. Otherwise, a negative value is returned.

**9.2.3.20 `jas_stream_seek()`**

```
JAS_DLLEXPORT long jas_stream_seek (
    jas_stream_t * stream,
    long offset,
    int origin )
```

Set the current position within the stream.

**Parameters**

<i>stream</i>	A pointer to the stream for which to set the current position.
<i>offset</i>	The new position for the stream.
<i>origin</i>	The origin to which this new position is relative.

The origin can be `SEEK_CUR`, `SEEK_SET`, or `SEEK_END` in a similar fashion as the `fseek` function in the C standard library (and the `lseek` function in POSIX).

**Returns**

Upon success, the new stream position is returned. Upon failure, a negative value is returned.

**9.2.3.21 `jas_stream_setrwcoun`**

```
JAS_DLLEXPORT long jas_stream_setrwcoun (
    jas_stream_t * stream,
    long rw_count )
```

Set the read/write count for a stream.

**Parameters**

<i>stream</i>	A pointer to the stream whose read/write count is to be set.
<i>rw_count</i>	The new value for the read/write count.

**Returns**

The old value of the read/write count is returned. This operation cannot fail.

**Todo** TODO/FIXME: Should this macro evaluate to void?

**9.2.3.22 jas\_stream\_setrwlimit()**

```
JAS_DLLEXPORT long jas_stream_setrwlimit (
    jas_stream_t * stream,
    long rwlimit )
```

Set the read/write limit for a stream.

**Parameters**

<i>stream</i>	A pointer to the stream whose read/write limit is to be set.
<i>rwlimit</i>	The new value for the read/write limit.

A negative read/write limit is treated as if it were infinity (i.e., there is no read/write limit).

**Returns**

The old read/write limit is returned.

**9.2.3.23 jas\_stream\_tell()**

```
JAS_ATTRIBUTE_PURE JAS_DLLEXPORT long jas_stream_tell (
    jas_stream_t * stream )
```

Get the current position within the stream.

**Parameters**

<i>stream</i>	A pointer to the stream whose current position is to be queried.
---------------	--



The current position of the stream is returned.

#### Returns

Upon success, the current stream position is returned. If an error is encountered, a negative value is returned.

#### 9.2.3.24 `jas_stream_tmpfile()`

```
JAS_DLLEXPORT jas_stream_t* jas_stream_tmpfile (
    void )
```

Open a temporary file as a stream.

A temporary file is created and opened as a stream. The temporary file is deleted when closed via [jas\\_stream\\_close\(\)](#). Some operating systems provide a mechanism for ensuring that a file is removed when closed. Such functionality may be used by the implementation when available.

#### Returns

Upon success, a pointer to the opened stream is returned. Otherwise, a null pointer is returned.

#### 9.2.3.25 `jas_stream_ungetc()`

```
JAS_DLLEXPORT int jas_stream_ungetc (
    jas_stream_t * stream,
    int c )
```

Put a character back on a stream.

#### Parameters

<i>stream</i>	A pointer to the stream to which the character should be put back.
<i>c</i>	The character to put back.

The character *c* (which was presumably read previously from the stream *stream*) is put back on the stream (as if it had not yet been read). In other words, this function undoes the effect of [jas\\_stream\\_getc\(\)](#). It is unspecified what happens if the character put back was not the one originally read. The number of characters that can be pushed back onto the stream for subsequent reading is limited. Trying to push back too many characters on a stream will result in an error. The approximate limit is given by the value of `JAS_STREAM_MAXPUTBACK`.

#### Returns

Upon success, zero is returned. If the specified character cannot be pushed back, a negative value is returned.

### 9.2.3.26 `jas_stream_write()`

```
JAS_DLLEXPORT unsigned jas_stream_write (
    jas_stream_t * stream,
    const void * buffer,
    unsigned count )
```

Write characters from a buffer to a stream.

#### Parameters

<i>stream</i>	A pointer to the stream to which to write data.
<i>buffer</i>	A pointer to the start of the buffer.
<i>count</i>	A count of the number of characters to write.

If `count` is zero, the function has no effect (and therefore cannot fail). Otherwise, the function will attempt to write `count` characters from the buffer starting at `buffer` to the stream `stream`. The number of characters written can be less than `count` due to an I/O error or the read/write limit being exceeded.

#### Returns

Upon success, the number of characters successfully written is returned. If an error occurs, the value returned will be less than `count`. The [jas\\_stream\\_error\(\)](#) and `jas_stream_rwlimit()` function (TODO/CHECK: the latter of which does not currently exist?) can be used to distinguish between:

1. failure due to an I/O error
2. failure due to the read/write limit being exceeded

#### Warning

TODO/FIXME: The type of the count parameter should be `size_t`. The return type should be `size_t`.

## 9.3 Image

### Classes

- struct [jas\\_image\\_cmpt\\_t](#)  
*Image component class.*
- struct [jas\\_image\\_t](#)  
*Image class.*
- struct [jas\\_image\\_cmptparm\\_t](#)  
*Component parameters class.*
- struct [jas\\_image\\_fmtops\\_t](#)  
*Image format-dependent operations.*
- struct [jas\\_image\\_fmtinfo\\_t](#)  
*Image format information.*

### Macros

- #define [JAS\\_IMAGE\\_MAXFMTS](#) 32  
*The maximum number of image data formats supported.*
- #define [jas\\_image\\_width](#)(image) ((image)->brx\_ - (image)->tlx\_)  
*Get the width of the image in units of the image reference grid.*
- #define [jas\\_image\\_height](#)(image) ((image)->bry\_ - (image)->tly\_)  
*Get the height of the image in units of the image reference grid.*
- #define [jas\\_image\\_tlx](#)(image) ((image)->tlx\_)  
*Get the x-coordinate of the top-left corner of the image bounding box on the reference grid.*
- #define [jas\\_image\\_tly](#)(image) ((image)->tly\_)  
*Get the y-coordinate of the top-left corner of the image bounding box on the reference grid.*
- #define [jas\\_image\\_brx](#)(image) ((image)->brx\_)  
*Get the x-coordinate of the bottom-right corner of the image bounding box on the reference grid (plus one).*
- #define [jas\\_image\\_bry](#)(image) ((image)->bry\_)  
*Get the y-coordinate of the bottom-right corner of the image bounding box on the reference grid (plus one).*
- #define [jas\\_image\\_numcmpts](#)(image) ((image)->numcmpts\_)  
*Get the number of image components.*
- #define [jas\\_image\\_clrspc](#)(image) ((image)->clrspc\_)  
*Get the color model used by the image.*
- #define [jas\\_image\\_setclrspc](#)(image, clrspc) ((image)->clrspc\_ = (clrspc))  
*Set the color model for an image.*
- #define [jas\\_image\\_cmptwidth](#)(image, cmptno) ((image)->cmpts\_[cmptno]->width\_)  
*Get the width of a component.*
- #define [jas\\_image\\_cmptheight](#)(image, cmptno) ((image)->cmpts\_[cmptno]->height\_)  
*Get the height of a component.*
- #define [jas\\_image\\_cmptsgnd](#)(image, cmptno) ((image)->cmpts\_[cmptno]->sgnd\_)  
*Get the signedness of the sample data for a component.*
- #define [jas\\_image\\_cmptprec](#)(image, cmptno) ((image)->cmpts\_[cmptno]->prec\_)  
*Get the precision of the sample data for a component.*
- #define [jas\\_image\\_cmptstep](#)(image, cmptno) ((image)->cmpts\_[cmptno]->hstep\_)

- *Get the horizontal subsampling factor for a component.*  
• #define [jas\\_image\\_cmptvstep](#)(image, cmptno) ((image)->cmpts\_[cmptno]->vstep\_)
- *Get the vertical subsampling factor for a component.*  
• #define [jas\\_image\\_cmpttlx](#)(image, cmptno) ((image)->cmpts\_[cmptno]->tlx\_)
- *Get the x-coordinate of the top-left corner of a component.*  
• #define [jas\\_image\\_cmpttly](#)(image, cmptno) ((image)->cmpts\_[cmptno]->tly\_)
- *Get the y-coordinate of the top-left corner of a component.*  
• #define [jas\\_image\\_cmptbrx](#)(image, cmptno)
- *Get the x-coordinate of the bottom-right corner of a component (plus "one").*  
• #define [jas\\_image\\_cmptbry](#)(image, cmptno)
- *Get the y-coordinate of the bottom-right corner of a component (plus "one").*  
• #define [jas\\_image\\_cmprofn](#)(image) ((image)->cmprofn\_)
- *Get the color management profile of an image.*  
• #define [jas\\_image\\_setcmprofn](#)(image, cmprofn) ((image)->cmprofn\_ = cmprofn)
- *Set the color management profile for an image.*

## Typedefs

- typedef int\_fast32\_t [jas\\_image\\_coord\\_t](#)  
*Image coordinate.*
- typedef int\_fast16\_t [jas\\_image\\_colorspc\\_t](#)  
*Color space (e.g., RGB, YCbCr).*
- typedef int\_fast32\_t [jas\\_image\\_cmpttype\\_t](#)  
*Component type (e.g., color, opacity).*
- typedef int\_fast16\_t [jas\\_image\\_smpltype\\_t](#)  
*Component sample data format (e.g., real/integer, signedness, precision).*

## Functions

- JAS\_DLLEXPORT [jas\\_image\\_t](#) \* [jas\\_image\\_create](#) (unsigned numcmpts, const [jas\\_image\\_cmptparm\\_t](#) \*cmptparms, [jas\\_clrspc\\_t](#) clrspc)  
*Create an image.*
- JAS\_DLLEXPORT [jas\\_image\\_t](#) \* [jas\\_image\\_create0](#) (void)  
*Create an "empty" image.*
- JAS\_DLLEXPORT [jas\\_image\\_t](#) \* [jas\\_image\\_copy](#) ([jas\\_image\\_t](#) \*image)  
*Clone an image.*
- JAS\_DLLEXPORT void [jas\\_image\\_destroy](#) ([jas\\_image\\_t](#) \*image)  
*Deallocate any resources associated with an image.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT bool [jas\\_image\\_cmpt\\_domains\\_same](#) (const [jas\\_image\\_t](#) \*image)  
*Test if all components are specified at the same positions in space.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT uint\_fast32\_t [jas\\_image\\_rawsize](#) (const [jas\\_image\\_t](#) \*image)  
*Get the raw size of an image (i.e., the nominal size of the image without any compression).*
- JAS\_DLLEXPORT [jas\\_image\\_t](#) \* [jas\\_image\\_decode](#) ([jas\\_stream\\_t](#) \*in, int fmt, const char \*optstr)  
*Create an image from a stream in some specified format.*
- JAS\_DLLEXPORT int [jas\\_image\\_encode](#) ([jas\\_image\\_t](#) \*image, [jas\\_stream\\_t](#) \*out, int fmt, const char \*optstr)  
*Write an image to a stream in a specified format.*

- JAS\_DLLEXPORT int [jas\\_image\\_readcmpt](#) ([jas\\_image\\_t](#) \*image, unsigned cmptno, [jas\\_image\\_coord\\_t](#) x, [jas\\_image\\_coord\\_t](#) y, [jas\\_image\\_coord\\_t](#) width, [jas\\_image\\_coord\\_t](#) height, [jas\\_matrix\\_t](#) \*data)  
*Read a rectangular region of an image component.*
- JAS\_DLLEXPORT int [jas\\_image\\_writecmpt](#) ([jas\\_image\\_t](#) \*image, unsigned cmptno, [jas\\_image\\_coord\\_t](#) x, [jas\\_image\\_coord\\_t](#) y, [jas\\_image\\_coord\\_t](#) width, [jas\\_image\\_coord\\_t](#) height, const [jas\\_matrix\\_t](#) \*data)  
*Write a rectangular region of an image component.*
- JAS\_DLLEXPORT void [jas\\_image\\_delcmpt](#) ([jas\\_image\\_t](#) \*image, unsigned cmptno)  
*Delete a component from an image.*
- JAS\_DLLEXPORT int [jas\\_image\\_addcmpt](#) ([jas\\_image\\_t](#) \*image, int cmptno, const [jas\\_image\\_cmptparm\\_t](#) \*cmptparm)  
*Add a component to an image.*
- JAS\_DLLEXPORT int [jas\\_image\\_copycmpt](#) ([jas\\_image\\_t](#) \*dstimage, unsigned dstcmptno, [jas\\_image\\_t](#) \*srcimage, unsigned srccmptno)  
*Copy a component from one image to another.*
- JAS\_DLLEXPORT int [jas\\_image\\_depalettize](#) ([jas\\_image\\_t](#) \*image, unsigned cmptno, unsigned numlutents, const int\_fast32\_t \*lutents, unsigned dtype, unsigned newcmptno)  
*Depalettize an image.*
- JAS\_DLLEXPORT int [jas\\_image\\_readcmptsample](#) ([jas\\_image\\_t](#) \*image, unsigned cmptno, unsigned x, unsigned y)  
*Read a component sample for an image.*
- JAS\_DLLEXPORT void [jas\\_image\\_writecmptsample](#) ([jas\\_image\\_t](#) \*image, unsigned cmptno, unsigned x, unsigned y, int\_fast32\_t v)  
*Write a component sample for an image.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT int [jas\\_image\\_getcmptbytype](#) (const [jas\\_image\\_t](#) \*image, [jas\\_image\\_cmpttype\\_t](#) ctype)  
*Get an image component by its type.*
- JAS\_DLLEXPORT void [jas\\_image\\_clearfmts](#) (void)  
*Clear the table of image formats.*
- JAS\_DLLEXPORT int [jas\\_image\\_addfmt](#) (int id, const char \*name, const char \*ext, const char \*desc, const [jas\\_image\\_fmtops\\_t](#) \*ops)  
*Add entry to table of image formats.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT int [jas\\_image\\_strtofmt](#) (const char \*s)  
*Get the ID for the image format with the specified name.*
- JAS\_ATTRIBUTE\_CONST JAS\_DLLEXPORT const char \* [jas\\_image\\_fmtostr](#) (int fmt)  
*Get the name of the image format with the specified ID.*
- JAS\_ATTRIBUTE\_CONST JAS\_DLLEXPORT const [jas\\_image\\_fmtinfo\\_t](#) \* [jas\\_image\\_lookupfmtbyid](#) (int id)  
*Lookup image format information by the format ID.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT const [jas\\_image\\_fmtinfo\\_t](#) \* [jas\\_image\\_lookupfmtbyname](#) (const char \*name)  
*Lookup image format information by the format name.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT int [jas\\_image\\_fmtfromname](#) (const char \*filename)  
*Guess the format of an image file based on its name.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT int [jas\\_image\\_getfmt](#) ([jas\\_stream\\_t](#) \*in)  
*Get the format of image data in a stream.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT int [jas\\_image\\_ishomosamp](#) (const [jas\\_image\\_t](#) \*image)  
*???*
- JAS\_DLLEXPORT int [jas\\_image\\_sampcmpt](#) ([jas\\_image\\_t](#) \*image, unsigned cmptno, unsigned newcmptno, [jas\\_image\\_coord\\_t](#) ho, [jas\\_image\\_coord\\_t](#) vo, [jas\\_image\\_coord\\_t](#) hs, [jas\\_image\\_coord\\_t](#) vs, int sgnd, unsigned prec)

???

- JAS\_DLLEXPORT int `jas_image_writecmt2` (`jas_image_t` \*image, unsigned cmptno, `jas_image_coord_t` x, `jas_image_coord_t` y, `jas_image_coord_t` width, `jas_image_coord_t` height, const long \*buf)

*Write sample data in a component of an image.*

- JAS\_DLLEXPORT int `jas_image_readcmt2` (`jas_image_t` \*image, unsigned cmptno, `jas_image_coord_t` x, `jas_image_coord_t` y, `jas_image_coord_t` width, `jas_image_coord_t` height, long \*buf)

*Read sample data in a component of an image.*

- JAS\_DLLEXPORT `jas_image_t` \* `jas_image_chclrspc` (`jas_image_t` \*image, const `jas_cmprof_t` \*outprof, `jas_cmxf` ← `cmxf`form\_intent\_t intent)

*Change the color space for an image.*

- JAS\_DLLEXPORT void `jas_image_dump` (`jas_image_t` \*image, FILE \*out)

*Dump the information for an image (for debugging).*

### 9.3.1 Detailed Description

### 9.3.2 Macro Definition Documentation

#### 9.3.2.1 `jas_image_brx`

```
#define jas_image_brx(  
    image ) ((image)->brx_)
```

Get the x-coordinate of the bottom-right corner of the image bounding box on the reference grid (plus one).

#### 9.3.2.2 `jas_image_bry`

```
#define jas_image_bry(  
    image ) ((image)->bry_)
```

Get the y-coordinate of the bottom-right corner of the image bounding box on the reference grid (plus one).

#### 9.3.2.3 `jas_image_clrspc`

```
#define jas_image_clrspc(  
    image ) ((image)->clrspc_)
```

Get the color model used by the image.

#### 9.3.2.4 jas\_image\_cmprof

```
#define jas_image_cmprof(  
    image ) ((image)->cmprof_)
```

Get the color management profile of an image.

#### 9.3.2.5 jas\_image\_cmptbrx

```
#define jas_image_cmptbrx(  
    image,  
    cmptno )
```

**Value:**

```
((image)->cmpts_[cmptno]->tlx_ + (image)->cmpts_[cmptno]->width_ * \  
    (image)->cmpts_[cmptno]->hstep_)
```

Get the x-coordinate of the bottom-right corner of a component (plus "one").

#### 9.3.2.6 jas\_image\_cmptbry

```
#define jas_image_cmptbry(  
    image,  
    cmptno )
```

**Value:**

```
((image)->cmpts_[cmptno]->tly_ + (image)->cmpts_[cmptno]->height_ * \  
    (image)->cmpts_[cmptno]->vstep_)
```

Get the y-coordinate of the bottom-right corner of a component (plus "one").

#### 9.3.2.7 jas\_image\_cmptheight

```
#define jas_image_cmptheight(  
    image,  
    cmptno ) ((image)->cmpts_[cmptno]->height_)
```

Get the height of a component.

### 9.3.2.8 jas\_image\_cmptthstep

```
#define jas_image_cmptthstep(  
    image,  
    cmptno ) ((image)->cmpts_[cmptno]->hstep_)
```

Get the horizontal subsampling factor for a component.

### 9.3.2.9 jas\_image\_cmptprec

```
#define jas_image_cmptprec(  
    image,  
    cmptno ) ((image)->cmpts_[cmptno]->prec_)
```

Get the precision of the sample data for a component.

### 9.3.2.10 jas\_image\_cmptsgnd

```
#define jas_image_cmptsgnd(  
    image,  
    cmptno ) ((image)->cmpts_[cmptno]->sgnd_)
```

Get the signedness of the sample data for a component.

### 9.3.2.11 jas\_image\_cmpttlx

```
#define jas_image_cmpttlx(  
    image,  
    cmptno ) ((image)->cmpts_[cmptno]->tlx_)
```

Get the x-coordinate of the top-left corner of a component.

### 9.3.2.12 jas\_image\_cmpttly

```
#define jas_image_cmpttly(  
    image,  
    cmptno ) ((image)->cmpts_[cmptno]->tly_)
```

Get the y-coordinate of the top-left corner of a component.



#### 9.3.2.13 `jas_image_cmptvstep`

```
#define jas_image_cmptvstep(  
    image,  
    cmptno ) ((image)->cmpts_[cmptno]->vstep_)
```

Get the vertical subsampling factor for a component.

#### 9.3.2.14 `jas_image_cmptwidth`

```
#define jas_image_cmptwidth(  
    image,  
    cmptno ) ((image)->cmpts_[cmptno]->width_)
```

Get the width of a component.

#### 9.3.2.15 `jas_image_height`

```
#define jas_image_height(  
    image ) ((image)->bry_ - (image)->tly_)
```

Get the height of the image in units of the image reference grid.

#### 9.3.2.16 `JAS_IMAGE_MAXFMTS`

```
#define JAS_IMAGE_MAXFMTS 32
```

The maximum number of image data formats supported.

#### 9.3.2.17 `jas_image_numcmpts`

```
#define jas_image_numcmpts(  
    image ) ((image)->numcmpts_)
```

Get the number of image components.

### 9.3.2.18 `jas_image_setclrspc`

```
#define jas_image_setclrspc(  
    image,  
    clrspc ) ((image)->clrspc_ = (clrspc))
```

Set the color model for an image.

### 9.3.2.19 `jas_image_setcmprof`

```
#define jas_image_setcmprof(  
    image,  
    cmprof ) ((image)->cmprof_ = cmprof)
```

Set the color management profile for an image.

### 9.3.2.20 `jas_image_tlx`

```
#define jas_image_tlx(  
    image ) ((image)->tlx_)
```

Get the x-coordinate of the top-left corner of the image bounding box on the reference grid.

### 9.3.2.21 `jas_image_tly`

```
#define jas_image_tly(  
    image ) ((image)->tly_)
```

Get the y-coordinate of the top-left corner of the image bounding box on the reference grid.

### 9.3.2.22 `jas_image_width`

```
#define jas_image_width(  
    image ) ((image)->brx_ - (image)->tlx_)
```

Get the width of the image in units of the image reference grid.

### 9.3.3 Typedef Documentation

#### 9.3.3.1 `jas_image_cmpttype_t`

```
typedef int_fast32_t jas_image_cmpttype_t
```

Component type (e.g., color, opacity).

#### 9.3.3.2 `jas_image_colorspc_t`

```
typedef int_fast16_t jas_image_colorspc_t
```

Color space (e.g., RGB, YCbCr).

#### 9.3.3.3 `jas_image_coord_t`

```
typedef int_fast32_t jas_image_coord_t
```

Image coordinate.

#### 9.3.3.4 `jas_image_smpltype_t`

```
typedef int_fast16_t jas_image_smpltype_t
```

Component sample data format (e.g., real/integer, signedness, precision).

### 9.3.4 Function Documentation

#### 9.3.4.1 `jas_image_addcmpt()`

```
JAS_DLLEXPORT int jas_image_addcmpt (
    jas_image_t * image,
    int cmptno,
    const jas_image_cmptparm_t * cmptparm )
```

Add a component to an image.

#### 9.3.4.2 `jas_image_addfmt()`

```
JAS_DLLEXPORT int jas_image_addfmt (
    int id,
    const char * name,
    const char * ext,
    const char * desc,
    const jas_image_fmtops_t * ops )
```

Add entry to table of image formats.

#### 9.3.4.3 `jas_image_chclrspc()`

```
JAS_DLLEXPORT jas_image_t* jas_image_chclrspc (
    jas_image_t * image,
    const jas_cmprof_t * outprof,
    jas_cmxform_intent_t intent )
```

Change the color space for an image.

#### 9.3.4.4 `jas_image_clearfmts()`

```
JAS_DLLEXPORT void jas_image_clearfmts (
    void )
```

Clear the table of image formats.

#### 9.3.4.5 `jas_image_cmpt_domains_same()`

```
JAS_ATTRIBUTE_PURE JAS_DLLEXPORT bool jas_image_cmpt_domains_same (
    const jas_image_t * image )
```

Test if all components are specified at the same positions in space.

#### 9.3.4.6 `jas_image_copy()`

```
JAS_DLLEXPORT jas_image_t* jas_image_copy (
    jas_image_t * image )
```

Clone an image.

#### 9.3.4.7 `jas_image_copycmpt()`

```
JAS_DLLEXPORT int jas_image_copycmpt (
    jas_image_t * dstimage,
    unsigned dstcmptno,
    jas_image_t * srcimage,
    unsigned srccmptno )
```

Copy a component from one image to another.

#### 9.3.4.8 `jas_image_create()`

```
JAS_DLLEXPORT jas_image_t* jas_image_create (
    unsigned numcmpts,
    const jas_image_cmptparm_t * cmptparms,
    jas_clrspc_t clrspc )
```

Create an image.

#### 9.3.4.9 `jas_image_create0()`

```
JAS_DLLEXPORT jas_image_t* jas_image_create0 (
    void )
```

Create an "empty" image.

#### 9.3.4.10 `jas_image_decode()`

```
JAS_DLLEXPORT jas_image_t* jas_image_decode (
    jas_stream_t * in,
    int fmt,
    const char * optstr )
```

Create an image from a stream in some specified format.

#### 9.3.4.11 `jas_image_delcmt()`

```
JAS_DLLEXPORT void jas_image_delcmt (
    jas_image_t * image,
    unsigned cmptno )
```

Delete a component from an image.

#### 9.3.4.12 `jas_image_depalettize()`

```
JAS_DLLEXPORT int jas_image_depalettize (
    jas_image_t * image,
    unsigned cmptno,
    unsigned numlutents,
    const int_fast32_t * lutents,
    unsigned dtype,
    unsigned newcmptno )
```

Depalettize an image.

#### 9.3.4.13 `jas_image_destroy()`

```
JAS_DLLEXPORT void jas_image_destroy (
    jas_image_t * image )
```

Deallocate any resources associated with an image.

#### 9.3.4.14 `jas_image_dump()`

```
JAS_DLLEXPORT void jas_image_dump (
    jas_image_t * image,
    FILE * out )
```

Dump the information for an image (for debugging).

#### 9.3.4.15 `jas_image_encode()`

```
JAS_DLLEXPORT int jas_image_encode (
    jas_image_t * image,
    jas_stream_t * out,
    int fmt,
    const char * optstr )
```

Write an image to a stream in a specified format.

#### 9.3.4.16 `jas_image_fmtfromname()`

```
JAS_ATTRIBUTE_PURE JAS_DLLEXPORT int jas_image_fmtfromname (
    const char * filename )
```

Guess the format of an image file based on its name.

#### 9.3.4.17 `jas_image_fmtostr()`

```
JAS_ATTRIBUTE_CONST JAS_DLLEXPORT const char* jas_image_fmtostr (
    int fmt )
```

Get the name of the image format with the specified ID.

#### 9.3.4.18 `jas_image_getcmptbytype()`

```
JAS_ATTRIBUTE_PURE JAS_DLLEXPORT int jas_image_getcmptbytype (
    const jas_image_t * image,
    jas_image_cmpttype_t ctype )
```

Get an image component by its type.

#### 9.3.4.19 `jas_image_getfmt()`

```
JAS_ATTRIBUTE_PURE JAS_DLLEXPORT int jas_image_getfmt (
    jas_stream_t * in )
```

Get the format of image data in a stream.

#### 9.3.4.20 `jas_image_ishomosamp()`

```
JAS_ATTRIBUTE_PURE JAS_DLLEXPORT int jas_image_ishomosamp (
    const jas_image_t * image )
```

???

#### 9.3.4.21 `jas_image_lookupfmtbyid()`

```
JAS_ATTRIBUTE_CONST JAS_DLLEXPORT const jas_image_fmtinfo_t* jas_image_lookupfmtbyid (
    int id )
```

Lookup image format information by the format ID.

#### 9.3.4.22 `jas_image_lookupfmtbyname()`

```
JAS_ATTRIBUTE_PURE JAS_DLLEXPORT const jas_image_fmtinfo_t* jas_image_lookupfmtbyname (
    const char * name )
```

Lookup image format information by the format name.

#### 9.3.4.23 `jas_image_rawsize()`

```
JAS_ATTRIBUTE_PURE JAS_DLLEXPORT uint_fast32_t jas_image_rawsize (
    const jas_image_t * image )
```

Get the raw size of an image (i.e., the nominal size of the image without any compression).



#### 9.3.4.24 `jas_image_readcmt()`

```
JAS_DLLEXPORT int jas_image_readcmt (
    jas_image_t * image,
    unsigned cmptno,
    jas_image_coord_t x,
    jas_image_coord_t y,
    jas_image_coord_t width,
    jas_image_coord_t height,
    jas_matrix_t * data )
```

Read a rectangular region of an image component.

The position and size of the rectangular region to be read is specified relative to the component's coordinate system.

#### 9.3.4.25 `jas_image_readcmt2()`

```
JAS_DLLEXPORT int jas_image_readcmt2 (
    jas_image_t * image,
    unsigned cmptno,
    jas_image_coord_t x,
    jas_image_coord_t y,
    jas_image_coord_t width,
    jas_image_coord_t height,
    long * buf )
```

Read sample data in a component of an image.

#### 9.3.4.26 `jas_image_readcmptsample()`

```
JAS_DLLEXPORT int jas_image_readcmptsample (
    jas_image_t * image,
    unsigned cmptno,
    unsigned x,
    unsigned y )
```

Read a component sample for an image.

#### 9.3.4.27 `jas_image_sampcmpt()`

```
JAS_DLLEXPORT int jas_image_sampcmpt (
    jas_image_t * image,
    unsigned cmptno,
    unsigned newcmptno,
    jas_image_coord_t ho,
    jas_image_coord_t vo,
    jas_image_coord_t hs,
    jas_image_coord_t vs,
    int sgnd,
    unsigned prec )
```

???

#### 9.3.4.28 `jas_image_strtofmt()`

```
JAS_ATTRIBUTE_PURE JAS_DLLEXPORT int jas_image_strtofmt (
    const char * s )
```

Get the ID for the image format with the specified name.

#### 9.3.4.29 `jas_image_writecmpt()`

```
JAS_DLLEXPORT int jas_image_writecmpt (
    jas_image_t * image,
    unsigned cmptno,
    jas_image_coord_t x,
    jas_image_coord_t y,
    jas_image_coord_t width,
    jas_image_coord_t height,
    const jas_matrix_t * data )
```

Write a rectangular region of an image component.

#### 9.3.4.30 `jas_image_writecmpt2()`

```
JAS_DLLEXPORT int jas_image_writecmpt2 (
    jas_image_t * image,
    unsigned cmptno,
    jas_image_coord_t x,
    jas_image_coord_t y,
    jas_image_coord_t width,
    jas_image_coord_t height,
    const long * buf )
```

Write sample data in a component of an image.

#### 9.3.4.31 `jas_image_writemptsample()`

```
JAS_DLLEXPORT void jas_image_writemptsample (
    jas_image_t * image,
    unsigned cmptno,
    unsigned x,
    unsigned y,
    int_fast32_t v )
```

Write a component sample for an image.



## Chapter 10

# Class Documentation

### 10.1 `jas_image_cmpt_t` Struct Reference

Image component class.

```
#include <jas_image.h>
```

#### 10.1.1 Detailed Description

Image component class.

The documentation for this struct was generated from the following file:

- [jas\\_image.h](#)

### 10.2 `jas_image_cmptparm_t` Struct Reference

Component parameters class.

```
#include <jas_image.h>
```

#### 10.2.1 Detailed Description

Component parameters class.

This data type exists solely/mainly for the purposes of the `jas_image_create` function.

The documentation for this struct was generated from the following file:

- [jas\\_image.h](#)

## 10.3 `jas_image_fmtinfo_t` Struct Reference

Image format information.

```
#include <jas_image.h>
```

### 10.3.1 Detailed Description

Image format information.

The documentation for this struct was generated from the following file:

- [jas\\_image.h](#)

## 10.4 `jas_image_fmtops_t` Struct Reference

Image format-dependent operations.

```
#include <jas_image.h>
```

### 10.4.1 Detailed Description

Image format-dependent operations.

The documentation for this struct was generated from the following file:

- [jas\\_image.h](#)

## 10.5 `jas_image_t` Struct Reference

Image class.

```
#include <jas_image.h>
```

### 10.5.1 Detailed Description

Image class.

The documentation for this struct was generated from the following file:

- [jas\\_image.h](#)

## Chapter 11

# File Documentation

### 11.1 jas\_cm.h File Reference

JasPer Color Management.

```
#include <jasper/jas_config.h>
#include <jasper/jas_icc.h>
```

#### 11.1.1 Detailed Description

JasPer Color Management.

### 11.2 jas\_compiler.h File Reference

Compiler-related macros.

```
#include <jasper/jas_config.h>
```

#### 11.2.1 Detailed Description

Compiler-related macros.

### 11.3 jas\_debug.h File Reference

JasPer Debugging-Related Functionality.

```
#include <jasper/jas_config.h>
#include <stdio.h>
```

### 11.3.1 Detailed Description

JasPer Debugging-Related Functionality.

## 11.4 jas\_dll.h File Reference

Shared Library Macros.

```
#include <jasper/jas_config.h>
```

### 11.4.1 Detailed Description

Shared Library Macros.

## 11.5 jas\_fix.h File Reference

JasPer Fixed-Point Number Class.

```
#include <jasper/jas_config.h>
#include <jasper/jas_types.h>
```

### 11.5.1 Detailed Description

JasPer Fixed-Point Number Class.

## 11.6 jas\_getopt.h File Reference

Command Line Option Parsing Code.

```
#include <jasper/jas_config.h>
```

### 11.6.1 Detailed Description

Command Line Option Parsing Code.



## 11.7 jas\_icc.h File Reference

ICC Profile.

```
#include <jasper/jas_config.h>
#include <jasper/jas_types.h>
#include <jasper/jas_stream.h>
#include <stdio.h>
```

### 11.7.1 Detailed Description

ICC Profile.

## 11.8 jas\_image.h File Reference

JasPer Image Class.

```
#include <jasper/jas_config.h>
#include <jasper/jas_stream.h>
#include <jasper/jas_types.h>
#include <jasper/jas_seq.h>
#include <jasper/jas_cm.h>
#include <stdio.h>
```

### Classes

- struct [jas\\_image\\_cmpt\\_t](#)  
*Image component class.*
- struct [jas\\_image\\_t](#)  
*Image class.*
- struct [jas\\_image\\_cmptparm\\_t](#)  
*Component parameters class.*
- struct [jas\\_image\\_fmtops\\_t](#)  
*Image format-dependent operations.*
- struct [jas\\_image\\_fmtinfo\\_t](#)  
*Image format information.*

## Macros

- `#define JAS_IMAGE_MAXFMTS 32`  
*The maximum number of image data formats supported.*
- `#define jas_image_width(image) ((image)->brx_ - (image)->tlx_)`  
*Get the width of the image in units of the image reference grid.*
- `#define jas_image_height(image) ((image)->bry_ - (image)->tly_)`  
*Get the height of the image in units of the image reference grid.*
- `#define jas_image_tlx(image) ((image)->tlx_)`  
*Get the x-coordinate of the top-left corner of the image bounding box on the reference grid.*
- `#define jas_image_tly(image) ((image)->tly_)`  
*Get the y-coordinate of the top-left corner of the image bounding box on the reference grid.*
- `#define jas_image_brx(image) ((image)->brx_)`  
*Get the x-coordinate of the bottom-right corner of the image bounding box on the reference grid (plus one).*
- `#define jas_image_bry(image) ((image)->bry_)`  
*Get the y-coordinate of the bottom-right corner of the image bounding box on the reference grid (plus one).*
- `#define jas_image_numcmpts(image) ((image)->numcmpts_)`  
*Get the number of image components.*
- `#define jas_image_clrspc(image) ((image)->clrspc_)`  
*Get the color model used by the image.*
- `#define jas_image_setclrspc(image, clrspc) ((image)->clrspc_ = (clrspc))`  
*Set the color model for an image.*
- `#define jas_image_cmptwidth(image, cmptno) ((image)->cmpts_[cmptno]->width_)`  
*Get the width of a component.*
- `#define jas_image_cmptheight(image, cmptno) ((image)->cmpts_[cmptno]->height_)`  
*Get the height of a component.*
- `#define jas_image_cmptsgnd(image, cmptno) ((image)->cmpts_[cmptno]->sgnd_)`  
*Get the signedness of the sample data for a component.*
- `#define jas_image_cmptprec(image, cmptno) ((image)->cmpts_[cmptno]->prec_)`  
*Get the precision of the sample data for a component.*
- `#define jas_image_cmptstep(image, cmptno) ((image)->cmpts_[cmptno]->hstep_)`  
*Get the horizontal subsampling factor for a component.*
- `#define jas_image_cmptvstep(image, cmptno) ((image)->cmpts_[cmptno]->vstep_)`  
*Get the vertical subsampling factor for a component.*
- `#define jas_image_cmpttlx(image, cmptno) ((image)->cmpts_[cmptno]->tlx_)`  
*Get the x-coordinate of the top-left corner of a component.*
- `#define jas_image_cmpttly(image, cmptno) ((image)->cmpts_[cmptno]->tly_)`  
*Get the y-coordinate of the top-left corner of a component.*
- `#define jas_image_cmptbrx(image, cmptno)`  
*Get the x-coordinate of the bottom-right corner of a component (plus "one").*
- `#define jas_image_cmptbry(image, cmptno)`  
*Get the y-coordinate of the bottom-right corner of a component (plus "one").*
- `#define jas_image_cmprofn(image) ((image)->cmprofn_)`  
*Get the color management profile of an image.*
- `#define jas_image_setcmprofn(image, cmprofn) ((image)->cmprofn_ = cmprofn)`  
*Set the color management profile for an image.*

## Typedefs

- typedef int\_fast32\_t [jas\\_image\\_coord\\_t](#)  
*Image coordinate.*
- typedef int\_fast16\_t [jas\\_image\\_colorspc\\_t](#)  
*Color space (e.g., RGB, YCbCr).*
- typedef int\_fast32\_t [jas\\_image\\_cmpttype\\_t](#)  
*Component type (e.g., color, opacity).*
- typedef int\_fast16\_t [jas\\_image\\_smplttype\\_t](#)  
*Component sample data format (e.g., real/integer, signedness, precision).*

## Functions

- JAS\_DLLEXPORT [jas\\_image\\_t](#) \* [jas\\_image\\_create](#) (unsigned numcmpts, const [jas\\_image\\_cmptparm\\_t](#) \*cmptparms, [jas\\_clrspc\\_t](#) clrspc)  
*Create an image.*
- JAS\_DLLEXPORT [jas\\_image\\_t](#) \* [jas\\_image\\_create0](#) (void)  
*Create an "empty" image.*
- JAS\_DLLEXPORT [jas\\_image\\_t](#) \* [jas\\_image\\_copy](#) ([jas\\_image\\_t](#) \*image)  
*Clone an image.*
- JAS\_DLLEXPORT void [jas\\_image\\_destroy](#) ([jas\\_image\\_t](#) \*image)  
*Deallocate any resources associated with an image.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT bool [jas\\_image\\_cmpt\\_domains\\_same](#) (const [jas\\_image\\_t](#) \*image)  
*Test if all components are specified at the same positions in space.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT uint\_fast32\_t [jas\\_image\\_rawsize](#) (const [jas\\_image\\_t](#) \*image)  
*Get the raw size of an image (i.e., the nominal size of the image without any compression).*
- JAS\_DLLEXPORT [jas\\_image\\_t](#) \* [jas\\_image\\_decode](#) ([jas\\_stream\\_t](#) \*in, int fmt, const char \*optstr)  
*Create an image from a stream in some specified format.*
- JAS\_DLLEXPORT int [jas\\_image\\_encode](#) ([jas\\_image\\_t](#) \*image, [jas\\_stream\\_t](#) \*out, int fmt, const char \*optstr)  
*Write an image to a stream in a specified format.*
- JAS\_DLLEXPORT int [jas\\_image\\_readcmpt](#) ([jas\\_image\\_t](#) \*image, unsigned cmptno, [jas\\_image\\_coord\\_t](#) x, [jas\\_image\\_coord\\_t](#) y, [jas\\_image\\_coord\\_t](#) width, [jas\\_image\\_coord\\_t](#) height, [jas\\_matrix\\_t](#) \*data)  
*Read a rectangular region of an image component.*
- JAS\_DLLEXPORT int [jas\\_image\\_writecmpt](#) ([jas\\_image\\_t](#) \*image, unsigned cmptno, [jas\\_image\\_coord\\_t](#) x, [jas\\_image\\_coord\\_t](#) y, [jas\\_image\\_coord\\_t](#) width, [jas\\_image\\_coord\\_t](#) height, const [jas\\_matrix\\_t](#) \*data)  
*Write a rectangular region of an image component.*
- JAS\_DLLEXPORT void [jas\\_image\\_delcmpt](#) ([jas\\_image\\_t](#) \*image, unsigned cmptno)  
*Delete a component from an image.*
- JAS\_DLLEXPORT int [jas\\_image\\_addcmpt](#) ([jas\\_image\\_t](#) \*image, int cmptno, const [jas\\_image\\_cmptparm\\_t](#) \*cmptparm)  
*Add a component to an image.*
- JAS\_DLLEXPORT int [jas\\_image\\_copycmpt](#) ([jas\\_image\\_t](#) \*dstimage, unsigned dstcmptno, [jas\\_image\\_t](#) \*srcimage, unsigned srccmptno)  
*Copy a component from one image to another.*
- JAS\_DLLEXPORT int [jas\\_image\\_depalettize](#) ([jas\\_image\\_t](#) \*image, unsigned cmptno, unsigned numlutents, const int\_fast32\_t \*lutents, unsigned dtype, unsigned newcmptno)  
*Depalettize an image.*

- JAS\_DLLEXPORT int [jas\\_image\\_readcmptsample](#) ([jas\\_image\\_t](#) \*image, unsigned cmptno, unsigned x, unsigned y)  
*Read a component sample for an image.*
- JAS\_DLLEXPORT void [jas\\_image\\_writecmptsample](#) ([jas\\_image\\_t](#) \*image, unsigned cmptno, unsigned x, unsigned y, int\_fast32\_t v)  
*Write a component sample for an image.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT int [jas\\_image\\_getcmptbytype](#) (const [jas\\_image\\_t](#) \*image, [jas\\_image\\_cmpttype\\_t](#) ctype)  
*Get an image component by its type.*
- JAS\_DLLEXPORT void [jas\\_image\\_clearfmts](#) (void)  
*Clear the table of image formats.*
- JAS\_DLLEXPORT int [jas\\_image\\_addfmt](#) (int id, const char \*name, const char \*ext, const char \*desc, const [jas\\_image\\_fmtops\\_t](#) \*ops)  
*Add entry to table of image formats.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT int [jas\\_image\\_strtofmt](#) (const char \*s)  
*Get the ID for the image format with the specified name.*
- JAS\_ATTRIBUTE\_CONST JAS\_DLLEXPORT const char \* [jas\\_image\\_fmtostr](#) (int fmt)  
*Get the name of the image format with the specified ID.*
- JAS\_ATTRIBUTE\_CONST JAS\_DLLEXPORT const [jas\\_image\\_fmtinfo\\_t](#) \* [jas\\_image\\_lookupfmtbyid](#) (int id)  
*Lookup image format information by the format ID.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT const [jas\\_image\\_fmtinfo\\_t](#) \* [jas\\_image\\_lookupfmtbyname](#) (const char \*name)  
*Lookup image format information by the format name.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT int [jas\\_image\\_fmtfromname](#) (const char \*filename)  
*Guess the format of an image file based on its name.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT int [jas\\_image\\_getfmt](#) ([jas\\_stream\\_t](#) \*in)  
*Get the format of image data in a stream.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT int [jas\\_image\\_ishomosamp](#) (const [jas\\_image\\_t](#) \*image)  
 ???
- JAS\_DLLEXPORT int [jas\\_image\\_sampcmpt](#) ([jas\\_image\\_t](#) \*image, unsigned cmptno, unsigned newcmptno, [jas\\_image\\_coord\\_t](#) ho, [jas\\_image\\_coord\\_t](#) vo, [jas\\_image\\_coord\\_t](#) hs, [jas\\_image\\_coord\\_t](#) vs, int sgnd, unsigned prec)  
 ???
- JAS\_DLLEXPORT int [jas\\_image\\_writecmpt2](#) ([jas\\_image\\_t](#) \*image, unsigned cmptno, [jas\\_image\\_coord\\_t](#) x, [jas\\_image\\_coord\\_t](#) y, [jas\\_image\\_coord\\_t](#) width, [jas\\_image\\_coord\\_t](#) height, const long \*buf)  
*Write sample data in a component of an image.*
- JAS\_DLLEXPORT int [jas\\_image\\_readcmpt2](#) ([jas\\_image\\_t](#) \*image, unsigned cmptno, [jas\\_image\\_coord\\_t](#) x, [jas\\_image\\_coord\\_t](#) y, [jas\\_image\\_coord\\_t](#) width, [jas\\_image\\_coord\\_t](#) height, long \*buf)  
*Read sample data in a component of an image.*
- JAS\_DLLEXPORT [jas\\_image\\_t](#) \* [jas\\_image\\_chclrspc](#) ([jas\\_image\\_t](#) \*image, const [jas\\_cmprof\\_t](#) \*outprof, [jas\\_cmxfm\\_intent\\_t](#) intent)  
*Change the color space for an image.*
- JAS\_DLLEXPORT void [jas\\_image\\_dump](#) ([jas\\_image\\_t](#) \*image, FILE \*out)  
*Dump the information for an image (for debugging).*

## 11.8.1 Detailed Description

JasPer Image Class.

## 11.9 jas\_init.h File Reference

JasPer Initialization/Cleanup Code.

```
#include <jasper/jas_config.h>
```

### Functions

- JAS\_DLLEXPORT int [jas\\_init](#) (void)  
*Initialize the JasPer library.*
- JAS\_DLLEXPORT void [jas\\_cleanup](#) (void)  
*Perform any clean up for the JasPer library.*

### 11.9.1 Detailed Description

JasPer Initialization/Cleanup Code.

## 11.10 jas\_malloc.h File Reference

JasPer Memory Allocator.

```
#include <jasper/jas_config.h>  
#include <stdio.h>
```

### 11.10.1 Detailed Description

JasPer Memory Allocator.

## 11.11 jas\_math.h File Reference

Math-Related Code.

```
#include <jasper/jas_config.h>  
#include <jasper/jas_compiler.h>  
#include <jasper/jas_types.h>  
#include <assert.h>  
#include <string.h>  
#include <stdint.h>  
#include <limits.h>
```

### 11.11.1 Detailed Description

Math-Related Code.

## 11.12 `jas_seq.h` File Reference

Sequence/Matrix Library.

```
#include <jasper/jas_config.h>
#include <jasper/jas_types.h>
#include <jasper/jas_math.h>
#include <stdio.h>
```

### 11.12.1 Detailed Description

Sequence/Matrix Library.

## 11.13 `jas_stream.h` File Reference

I/O Stream Class.

```
#include <jasper/jas_config.h>
#include <stdio.h>
#include <jasper/jas_types.h>
```

### Macros

- `#define jas_stream_eof(stream) (((stream)->flags_ & JAS_STREAM_EOF) != 0)`  
*Get the EOF indicator for a stream.*
- `#define jas_stream_error(stream) (((stream)->flags_ & JAS_STREAM_ERR) != 0)`  
*Get the error indicator for a stream.*
- `#define jas_stream_clearerr(stream) ((stream)->flags_ &= ~(JAS_STREAM_ERR | JAS_STREAM_EOF))`  
*Clear the error indicator for a stream.*
- `#define jas_stream_getrwlmit(stream) (((const jas_stream_t*)(stream))->rwlmit_)`  
*Get the read/write limit for a stream.*
- `#define jas_stream_getrwcoun(stream) (((const jas_stream_t*)(stream))->rwcnt_)`  
*Get the read/write count for a stream.*
- `#define jas_stream_getc(stream) jas_stream_getc_func(stream)`  
*jas\_stream\_getc Read a character from a stream.*
- `#define jas_stream_putc(stream, c) jas_stream_putc_func(stream, c)`  
*jas\_stream\_putc Write a character to a stream.*
- `#define jas_stream_peekc(stream)`  
*Look at the next character to be read from a stream without actually removing the character from the stream.*

## Functions

- JAS\_DLLEXPORT jas\_stream\_t \* [jas\\_stream\\_fopen](#) (const char \*filename, const char \*mode)  
*Open a file as a stream.*
- JAS\_DLLEXPORT jas\_stream\_t \* [jas\\_stream\\_memopen](#) (char \*buffer, int buffer\_size)  
*Open a memory buffer as a stream.*
- JAS\_DLLEXPORT jas\_stream\_t \* [jas\\_stream\\_memopen2](#) (char \*buffer, size\_t buffer\_size)
- JAS\_DLLEXPORT jas\_stream\_t \* [jas\\_stream\\_fdopen](#) (int fd, const char \*mode)  
*Open a file descriptor as a stream.*
- JAS\_DLLEXPORT jas\_stream\_t \* [jas\\_stream\\_freopen](#) (const char \*path, const char \*mode, FILE \*fp)  
*Open a stdio (i.e., C standard library) stream as a stream.*
- JAS\_DLLEXPORT jas\_stream\_t \* [jas\\_stream\\_tmpfile](#) (void)  
*Open a temporary file as a stream.*
- JAS\_DLLEXPORT int [jas\\_stream\\_close](#) (jas\_stream\_t \*stream)  
*Close a stream.*
- JAS\_DLLEXPORT long [jas\\_stream\\_setrwlimit](#) (jas\_stream\_t \*stream, long rwlimit)  
*Set the read/write limit for a stream.*
- JAS\_DLLEXPORT long [jas\\_stream\\_setrwcoun](#) (jas\_stream\_t \*stream, long rw\_count)  
*Set the read/write count for a stream.*
- JAS\_DLLEXPORT unsigned [jas\\_stream\\_read](#) (jas\_stream\_t \*stream, void \*buffer, unsigned count)  
*Read characters from a stream into a buffer.*
- JAS\_DLLEXPORT unsigned [jas\\_stream\\_peek](#) (jas\_stream\_t \*stream, void \*buffer, size\_t count)  
*Attempt to retrieve one or more pending characters of input from a stream into a buffer without actually removing the characters from the stream.*
- JAS\_DLLEXPORT unsigned [jas\\_stream\\_write](#) (jas\_stream\_t \*stream, const void \*buffer, unsigned count)  
*Write characters from a buffer to a stream.*
- JAS\_DLLEXPORT int [jas\\_stream\\_printf](#) (jas\_stream\_t \*stream, const char \*format,...)  
*Write formatted output to a stream.*
- JAS\_DLLEXPORT int [jas\\_stream\\_puts](#) (jas\_stream\_t \*stream, const char \*s)  
*Write a string to a stream.*
- JAS\_DLLEXPORT char \* [jas\\_stream\\_gets](#) (jas\_stream\_t \*stream, char \*buffer, int buffer\_size)  
*Read a line of input from a stream.*
- JAS\_DLLEXPORT int [jas\\_stream\\_ungetc](#) (jas\_stream\_t \*stream, int c)  
*Put a character back on a stream.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT int [jas\\_stream\\_isseekable](#) (jas\_stream\_t \*stream)  
*Determine if stream supports seeking.*
- JAS\_DLLEXPORT long [jas\\_stream\\_seek](#) (jas\_stream\_t \*stream, long offset, int origin)  
*Set the current position within the stream.*
- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT long [jas\\_stream\\_tell](#) (jas\_stream\_t \*stream)  
*Get the current position within the stream.*
- JAS\_DLLEXPORT int [jas\\_stream\\_rewind](#) (jas\_stream\_t \*stream)  
*Seek to the beginning of a stream.*
- JAS\_DLLEXPORT int [jas\\_stream\\_flush](#) (jas\_stream\_t \*stream)  
*Flush any pending output to a stream.*
- JAS\_DLLEXPORT int [jas\\_stream\\_copy](#) (jas\_stream\_t \*destination, jas\_stream\_t \*source, int count)  
*Copy data from one stream to another.*
- JAS\_DLLEXPORT int [jas\\_stream\\_display](#) (jas\_stream\_t \*stream, FILE \*fp, int count)

*Print a hex dump of data read from a stream.*

- JAS\_DLLEXPORT int [jas\\_stream\\_gobble](#) (jas\_stream\_t \*stream, int count)

*Consume (i.e., discard) characters from stream.*

- JAS\_DLLEXPORT int [jas\\_stream\\_pad](#) (jas\_stream\_t \*stream, int count, int value)

*Write a fill character multiple times to a stream.*

- JAS\_ATTRIBUTE\_PURE JAS\_DLLEXPORT long [jas\\_stream\\_length](#) (jas\_stream\_t \*stream)

*Get the size of the file associated with the specified stream.*

### 11.13.1 Detailed Description

I/O Stream Class.

## 11.14 [jas\\_string.h](#) File Reference

String Library.

```
#include <jasper/jas_config.h>
```

### 11.14.1 Detailed Description

String Library.

## 11.15 [jas\\_tmr.h](#) File Reference

Timer Code.

```
#include <jasper/jas_config.h>
#include <time.h>
```

### 11.15.1 Detailed Description

Timer Code.

## 11.16 [jas\\_tvp.h](#) File Reference

Tag/Value Pair Parser.

```
#include <jasper/jas_config.h>
```



### 11.16.1 Detailed Description

Tag/Value Pair Parser.

## 11.17 jas\_types.h File Reference

Primitive Types.

```
#include <jasper/jas_config.h>
#include <stddef.h>
#include <stdint.h>
#include <stdbool.h>
#include <inttypes.h>
```

### 11.17.1 Detailed Description

Primitive Types.

## 11.18 jas\_version.h File Reference

JasPer Version.

```
#include <jasper/jas_config.h>
```

### 11.18.1 Detailed Description

JasPer Version.

## 11.19 jasper.h File Reference

JasPer Main Header.

```
#include <jasper/jas_config.h>
#include <jasper/jas_types.h>
#include <jasper/jas_version.h>
#include <jasper/jas_init.h>
#include <jasper/jas_cm.h>
#include <jasper/jas_icc.h>
#include <jasper/jas_fix.h>
#include <jasper/jas_debug.h>
```

```
#include <jasper/jas_getopt.h>
#include <jasper/jas_image.h>
#include <jasper/jas_math.h>
#include <jasper/jas_malloc.h>
#include <jasper/jas_seq.h>
#include <jasper/jas_stream.h>
#include <jasper/jas_string.h>
#include <jasper/jas_tmr.h>
#include <jasper/jas_tvp.h>
```

### 11.19.1 Detailed Description

JasPer Main Header.