

JasPer
4.2.9

Generated by Doxygen 1.16.1

Chapter 1

Overview

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 Features

Some of the features of the JasPer library are as follows:

- has multithreading support (e.g., the JasPer library can be used concurrently in multiple threads)
- has native support for several image codecs, including:
 - JPEG-2000 JP2 File Format Syntax (ISO/IEC 15444-1)
 - JPEG-2000 Code Stream Syntax (ISO/IEC 15444-1)
 - Portable Graymap/Pixmap (PNM)
 - Microsoft Bitmap (BMP)
 - Sun Rasterfile (RAS)
 - JPEG-2000 VM Format (PGX)
- has non-native support for the following image codecs:
 - JPEG (ISO/IEC 10918-1) via IJG JPEG Library
 - HEIC (ISO/IEC 23008-12) via the libheif Library
- can add new codecs (or enable/disable codecs) at run time
- can specify a memory allocator to be used by the library
- can specify a logging function to be used by the library for error, warning, informational, and debugging messages

- can place an upper limit on the total amount of memory that can be used by the library (which is useful in protecting against malicious image streams during decoding)
- can decode from a non-seekable source stream (which is useful for filtering pipelines)
- can manage ICC profiles
- portable code, known to work on many platforms (e.g., Linux, Windows, MacOS, BSD, etc.)
- can autodetect the image format during decoding

Several application programs are also provided:

- `jasper`, an image transcoder program for converting between image formats
- `imginfo`, a program for querying the properties of an image, such as: width, height, number of components, and the number of bits per sample
- `imgcmp`, a program for comparing two images using various distance metrics, such as peak absolute error (PAE), mean absolute error (MAE), and peak signal to noise ratio (PSNR)
- `jiv`, an image viewer

1.3 News

In release 3.0.0 of JasPer, the following notable changes were made to the API and/or behavior of the library relative to earlier releases (and therefore impact backward compatibility):

- The `jas_init()` and `jas_cleanup()` functions have been marked as deprecated. The replacements for these functions are described in [Configuration, Initialization, and Shutdown](#).
- The `jas_init()` function no longer registers `jas_cleanup()` as an atexit handler. So, the library user is now responsible for invoking `jas_cleanup()` at an appropriate time. This change was needed in order to better support the use of the JasPer library on some platforms.
- In the I/O streams API, some of the integral types used in the parameter/return types of the following functions were changed to use `size_t/ssize_t` in order to address some longstanding API issues:
 - `jas_stream_memopen()`
 - `jas_stream_read()`
 - `jas_stream_write()`
 - `jas_stream_copy()`
 - `jas_stream_gobble()`
 - `jas_stream_copy2()`
 - `jas_stream_pad()`
- In the I/O streams API, in some cases where the integral type used was changed from a signed to unsigned type, deprecated behavior that allowed for negative values was removed.
- The `jas_stream_memopen2()` function has been deprecated.

- More effort has been made to hide functions/macros that are internal to the JasPer library in an effort to prevent applications using such functions/macros (which can lead to many types of problems).

As part of the work done in preparing the JasPer 3.0.0 release, support for JasPer 3.0.0 was merged into the popular XV software, which can be obtained at:

- <https://github.com/jasper-software/xv.git>

4.2.9 (2026-03-05)
=====

- * Fixed a bug in the JP2 encoder that caused incorrect handling of opacity components in some cases.

4.2.8 (2025-08-05)
=====

- * Fixed a bug in the JPC decoder that could cause bad memory accesses if the debug level is set sufficiently high (#402, #403) (CVE-2025-8837).

4.2.7 (2025-08-02)
=====

- * Added some missing range checking on several coding parameters in the JPC encoder (#401) (CVE-2025-8836).

4.2.6 (2025-08-02)
=====

- * Added a check for a missing color component in the `jas_image_chclrspc` function (#400) (CVE-2025-8835).
- * Fixed a minor build problem related to the use of `-Wstrict-prototypes` with Clang.

4.2.5 (2025-03-26)
=====

- * Made a change to a configuration header file in order to avoid undesirable compiler warnings when JasPer is used in C++ code (#393).

4.2.4 (2024-04-27)
=====

- * Added some missing checks to the `jas_heic_decode` function in the HEIC codec (#383).

4.2.3 (2024-03-30)
=====

- * Added a missing check in the JPC codec (#381) (CVE-2024-31744).

4.2.2 (2024-03-11)
=====

- * Fix minor build issue (#374).

4.2.1 (2024-02-18)
=====

- * Fix a build problem for the DJGPP/MS-DOS environment (#372).

4.2.0 (2024-02-05)
=====

- * Add the JAS_PACKAGING option to the CMake build in an attempt to allow easier control over rpath settings by packagers of JasPer.
- * Remove a number of obsolete scripts.
- * Make some cosmetic changes to the code for the JPC codec in order to improve readability (#371).
- * Fix a portability bug related to threads/atomics.
- * Replace some lingering uses of strtok in the JPC coder with jas_strtok, since the use of strtok is problematic in multithreading contexts.

4.1.2 (2024-01-11)
=====

- * Fix invalid memory write bug (#367) (CVE-2023-51257).
- * Fix missing range check in the JPC encoder (#368).

4.1.1 (2023-11-28)
=====

- * Disallow in-source builds by default.
- * Fix a potential integer overflow problem in the jas_get_total_mem_size function (for the Windows platform).

4.1.0 (2023-11-04)
=====

- * Add support for building several JasPer application programs for WebAssembly target with WASI support.

4.0.1 (2023-11-04)
=====

- * Fix integer overflow bug in PNM decoder (#353).
- * Fix a few minor build issues.

4.0.0 (2022-11-05)
=====

- * Improve static linking (#336).
- * Fix path relocation in mingw environment (#335).
- * Improve logging and build scripts.
- * Improve JPEG-2000 conformance test results.
- * Enable PIC by default.
- * Fix memory leaks in function cmdopts_parse (#332) (CVE-2022-2963).
- * imgcmp:
 - + Add quiet (-q) option.
 - + Add debug-level option.
 - + Fix memory leak.
- * imginfo:
 - + Add quiet (-q) option.
- * Fix bug in parsing PGX header.
- * Fix integer overflow bug (#345) (CVE-2022-40755).

3.0.6 (2022-07-13)
=====

- * Fix bug in manual deployment script.

3.0.5 (2022-06-23)
=====

- * Fix a minor build issue (#328).

3.0.4 (2022-06-02)
=====

- * Eliminate some bogus calls to abort.
- * Fix a typo in jas_safeui64_div (#323).

- * Add some additional logging messages.
- * Fix the source of a potential compiler warning (#321).

3.0.3 (2022-03-15)
=====

- * Fix some portability issues in a few scripts.

3.0.2 (2022-02-14)
=====

- * Fix a build issue that occurs when a cross-compiler is used (e.g., #319).

3.0.1 (2022-02-12)
=====

- * Fix some build/portability issues (e.g., #317, #318).

3.0.0 (2022-02-05)
=====

VERY IMPORTANT NOTE:

This release of the Jasper software introduced some changes in the API and/or behavior of the library relative to earlier releases, which may necessitate some small changes in code using the library (e.g., to avoid memory leaks or other problems). Please refer to the "News" section of the Jasper Reference Manual for more details. For convenience, this manual is available online (for various Jasper releases) at:
<https://jasper-software.github.io/jasper-manual>

- * Greatly improve documentation.
- * Add support for multithreading.
- * Add some customization points in the library, such as the memory allocator and error logging function.
- * Add improved memory usage tracking and limiting.
- * Add experimental partial encoding/decoding support for the HEIC format.
- * Fix some longstanding issues in the Jasper I/O streams API.
- * Add the running of the full test suite in CI builds for the Windows platform. (Previously, the full test suite was only run for CI builds on Unix-based platforms.)
- * Fix many bugs (e.g., #305, #307, #308, #309, #312, #314, and many others not associated with any issue numbers).

- * Merged support for Jasper 3.0.0 into the XV software at:
<https://github.com/jasper-software/xv.git>

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

- * Fix a JP2/JPC decoder bug. (#291)
- * Fix a build issue impacting some platforms. (#296)

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) (CVE-2021-3443)

- * 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-2021-27845
<https://github.com/mdadams/jasper/issues/194> (part 1)
 - * 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_DTYPETOBPC, 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

```

Chapter 2

License

2.1 License

The JasPer software may only be used under the terms of the license below. A copy of the license can also be found in the file named "LICENSE.txt" in the top-level directory of the software distribution for JasPer.

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.

Chapter 3

Introduction

3.1 Motivation Behind JasPer

Digital imagery is used in many of today's computer software applications. Consequently, software modules that facilitate the handling of such data are often needed. Almost any application program that deals with images must address the problem of image interchange and import/export. That is, a means must exist for moving image data between an application and its external environment. Moreover, applications must often be capable of rendering an image for display on a particular output device (such as a monitor or printer) with reasonably accurate color/tone reproduction.

Although image import/export and rendering are very basic functionalities, they are often not easily implemented. Usually, an image is represented in a coded format (such as JPEG-2000 JP2 [?] or JPEG [?]). Since coding formats are frequently quite complex, the import/export of image data can be a daunting task. Rendering an image in such a way as to accurately reproduce color/tone requires a color management scheme of some sort. Unfortunately, developing an effective color management engine can require considerable effort.

A search for a solution to the above problems led to the development of the JasPer software. The remainder of this document provides a detailed description of the this software.

3.2 What is JasPer?

In simple terms, JasPer is a software tool kit for the handling of image data. The software provides a means for representing images, and facilitates the manipulation of image data, as well as the import/export of such data in numerous formats (e.g., JPEG-2000 JP2 [?], JPEG [?], PNM [?], BMP [?], Sun Rasterfile [?], and PGX [?]). The import functionality supports the auto-detection (i.e., automatic determination) of the image format, eliminating the need to explicitly identify the format of coded input data. A simple color management engine is also provided in order to allow the accurate representation of color. Partial support is included for the ICC color profile file format [?], an industry standard for specifying color.

The JasPer software consists of a library and several application programs. The code is written in the C programming language [?] [?]. This language was chosen primarily due to the availability of C development environments for most of today's computing platforms. At present, JasPer consists of approximately 40K lines of code. Although written in C, the JasPer library can be easily integrated into applications written in the C++ programming language as well.

3.3 Software License

JasPer software development is based on an open-source model. A copy of the software license can be found in [License](#).

3.4 Other Sources of Information on JasPer

For more information about the JasPer software, please visit the following web pages:

- Official JasPer Project Home Page, <https://jasper-software.github.io/jasper>
- JasPer GitHub Home Page, <https://github.com/jasper-software/jasper>
- JasPer Reference Manual: <https://jasper-software.github.io/jasper-manual>
- Michael Adams' JasPer Page, <https://www.ece.uvic.ca/~mdadams/jasper>

For more information about the JasPer software and JPEG-2000 standard, the reader is referred to [?] [?] .

3.5 Origin of the Name

The JasPer software is named, in part, after Jasper National Park, the largest national park in the Canadian Rockies with 10,878 square kilometres of mountain wilderness. As it happens, jasper is also the name of an opaque cryptocrystalline variety of quartz used for ornamentation or as a gemstone—hence, the implication that the software is precious (i.e., like a gemstone). Lastly, the name "jasper" was also chosen because it contains a letter "J" followed subsequently by a letter "P", not unlike the abbreviation "JP" that is associated with the JPEG-2000 standard.

Chapter 4

Installation

4.1 Version Identification

As the Jasper software continues to evolve over time, it is important to be able to identify particular releases of the software. Every release of the Jasper software is named by a version identifier. A version identifier is comprised of three integers separated by dots. In order, the three integers correspond to the major, minor, and micro version numbers for the software. For example, the version identifier "1.500.0" corresponds to a major version of 1, a minor version of 500, and a micro version of 0. In instances where the micro version is zero, the version identifier may be truncated after the minor version number. For example, the version identifier "1.500" is completely valid and simply an abbreviation for "1.500.0".

Given two different releases of the Jasper software, one can easily determine which one is more recent by comparing the version identifiers, as follows:

1. if the major version numbers differ, the release with the higher major version number is newer;
2. if the major version numbers are equal and the minor version numbers differ, the release with the higher minor version number is newer; or
3. if the major version numbers are equal and the minor version numbers are equal, the release with the higher micro version is newer.

4.2 Obtaining the Software

The latest version of the Jasper software can be downloaded from the following locations:

- Jasper Project GitHub Releases Page, <https://github.com/jasper-software/jasper/releases> (The official releases of Jasper can be found here.)
- Jasper Project Home Page, <https://jasper-software.github.io/jasper>

4.3 Extracting the Software

The JasPer software is distributed in the form of a gzipped tar archive. Therefore, in order to extract the contents of this file, a program capable of handling gzipped tar archives is required. Such software is readily available for most modern computing platforms.

4.4 Prerequisites for Building (Software Dependencies)

The JasPer code should compile and run on any platform with a C language implementation conforming to ISO/IEC 9899:2011 [?] (i.e., the ISO C language standard) and supporting a subset of ISO/IEC 9945-1 [?] (i.e., the POSIX C API). Only limited POSIX support is required (e.g., the `open`, `close`, `read`, `write`, and `lseek` functions must be supported).

If you need a C compiler that is reasonably compliant with the ISO/IEC 9899:2011 standard, you can obtain GNU C from the GNU Project web site (i.e., <http://www.gnu.org>). If you need the CMake software, this can be obtained from the official CMake web site (i.e., <https://cmake.org>).

If you are unfortunate enough to have a compiler that is not compliant with ISO/IEC 9899:2011, you may need to make some changes to the code. Unfortunately, even some of the most popular C language implementations do not strictly comply with the standard. One such example is Microsoft Visual C 6.0. Due to the popularity of Visual C, however, several workarounds have been added to the JasPer code to ensure that it will compile successfully with Visual C.

Portability was a major consideration during the design of the JasPer software. For this reason, the software makes minimal assumptions about the runtime environment. The code uses very little floating-point arithmetic, most of which can be attributed to floating-point conversions in invocations of the `printf` function. This minimal use of floating-point arithmetic should make the code much easier to port to platforms lacking hardware support for floating-point arithmetic.

In order to have access to the full functionality of the JasPer software, you may need to install some additional software on your system. This software must be installed before you attempt to build JasPer.

In order to build the JasPer software with JPEG support, you will need to download and install the free IJG JPEG library which is available from the IJG web site:

- <http://www.ijg.org>

In order to build the JasPer software with HEIC support, the Libheif library is needed. This library is available from:

- <https://github.com/strukturag/libheif>

In order to build the jiv application, you will need the OpenGL and GLUT libraries installed on your system. Most modern systems include OpenGL support (including Linux, MacOS, and Microsoft Windows). The GLUT library is relatively less common and, therefore, may not be installed on your system. To obtain the GLUT library, one can visit:

- <http://freeglut.sourceforge.net>

For more information on the OpenGL library, see:

- <http://www.opengl.org>.

4.5 Building the Software

Obviously, before the software can be built, the contents of the archive file containing the Jasper distribution must be extracted.

The Jasper software is intended to be built using CMake, a very popular cross-platform build tool. The current version of the Jasper software should compile on most modern Unix variants such as Linux and MacOS as well as Microsoft Windows. The continuous integration (CI) framework on GitHub used for Jasper development includes builds for the following environments:

- the GCC and Clang C compilers on Ubuntu Linux
- the GCC and Clang C compiler on MacOS
- the Microsoft Visual C (MSVC) compiler on Microsoft Windows

Also, the lead Jasper developer uses both the GCC and Clang compilers on Fedora Linux for much of their work. So, the Jasper software should build fairly reliably in these environments. Of course, the software should compile successfully in many other environments as well.

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. To build the software, the following steps are required (in order):

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

2. Configure the software using CMake.
To do this, invoke the command:

```
cmake -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".

This process allows important information about the system configuration to be determined. Unless you know what you are doing (or have problems with the default build settings), it is **STRONGLY RECOMMENDED** that you not override the default build settings.

3. Build the software using CMake.
To do this, invoke the command:

```
cmake --build $BUILD_DIR
```

(Assuming that the build operation was successful, the executables for the Jasper application programs should be located in the directory `$BUILD_DIR/src/app`.)

4. Run the test suite to ensure that the software seems to be working

correctly. To do this, invoke the commands:

```
cd $BUILD_DIR
ctest --output-on-failure
```

5. Install the software (i.e., the library, application programs, header files, and other auxiliary data).

To do this, invoke the command:

```
cmake --build $BUILD_DIR --target install
```

This step may require special (e.g., administrator) privileges depending on the target directory for installation. (On Unix-based systems, the default installation directory is typically under `usr/local`.)

Assuming that the software was installed successfully, the executables for the Jasper application programs should be found somewhere under the `$CMAKE_INSTALL_PREFIX` directory (e.g., `$CMAKE_INSTALL_PREFIX/bin`).

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% ^
-DMAKE_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
```

Other Remarks

In some cases, it may be necessary to explicitly disable the use of the IJG JPEG library (i.e., `libjpeg`). This is accomplished through a CMake configuration option. (See the "CMake Options" section below.) For example, such action may be required if the version of the JPEG library installed on your system is not compatible with the version of Jasper being built. Also, when building under the Cygwin environment, it may be necessary to explicitly disable the use of the JPEG library.

In some situations, it may be necessary to explicitly disable the use of the OpenGL libraries. This is accomplished through a CMake configuration option.

(See the "CMake Options" section below.)

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_MULTITHREADING_SUPPORT
Specify if the library should include multithreading support.
Valid values: true and false

JAS_PREFER_PTHREAD
Specify if the POSIX Threads library should be preferred over native C11 threading support.
Valid values: true and false

JAS_ENABLE_PROGRAMS
Specify if the demo application programs should be built/installed.
Valid values: true and false

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_DEFAULT_MAX_MEM_USAGE
Specify the maximum amount of memory (in bytes) that may be used by the library.
This value is only a default and can be overridden at run time.
Valid value: a (strictly) positive integer

JAS_INCLUDE_BMP_CODEC
JAS_INCLUDE_JP2_CODEC
JAS_INCLUDE_JPC_CODEC
JAS_INCLUDE_JPG_CODEC
JAS_INCLUDE_MIF_CODEC

```
JAS_INCLUDE_PGX_CODEC
JAS_INCLUDE_PNM_CODEC
JAS_INCLUDE_RAS_CODEC
    Include support for the specified codec in the library.
    (This support can still be disabled at run time.)
    Valid values: true or false

JAS_ENABLE_BMP_CODEC
JAS_ENABLE_JP2_CODEC
JAS_ENABLE_JPC_CODEC
JAS_ENABLE_JPG_CODEC
JAS_ENABLE_MIF_CODEC
JAS_ENABLE_PGX_CODEC
JAS_ENABLE_PNM_CODEC
JAS_ENABLE_RAS_CODEC
    Enable support for the specified codec by default at run time.
    Valid values: true or false

JAS_ENABLE_ASAN
    Enable the Address Sanitizer.
    Valid values: true or false

JAS_ENABLE_UBSAN
    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
```

Chapter 5

JasPer Library

5.1 Introduction

The heart of the JasPer software is the JasPer library. In fact, most of the code in JasPer is associated with this library (as opposed to the JasPer sample application programs). The JasPer library provides classes for representing images, color profiles (i.e., color space definitions), and other related entities. Each of these classes has a well-defined interface through which an application may interact with class objects. The library can be used to manipulate images, import/export image data in a variety of formats, and perform basic color management operations.

Conceptually, the JasPer library is structured as shown in Figure [fig__swstruct](#). The library consists of two distinct types of code:

1. core code, and
2. codec drivers.

The core code provides the basic framework upon which the library is built, while the codec drivers only provide the means for encoding/decoding image data in various formats. All application interfaces are through the core code. The codec drivers are only ever directly called by the core code, never by an application.

The codec support in the JasPer library is both modular and extensible. A well-defined interface exists between the core code and codec drivers. Moreover, support for a new image format can be easily added without having to modify the library in any way. To do so, a codec driver for the new format simply needs to be provided. Furthermore, an application need only include codec drivers for the image formats that it will use. In this way, an application can avoid the cost of increased memory consumption for codec drivers that are never to be employed.

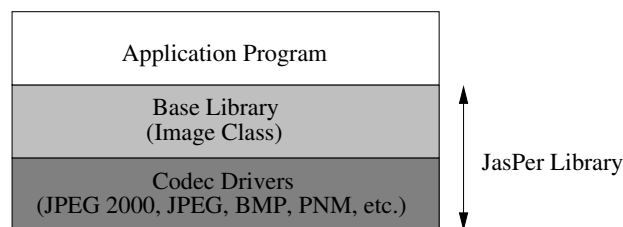


Figure 5.1 Software structure

5.2 Core Code

To avoid name space collisions, all of the identifiers used by the core code are prefixed with either `jas_` or `JAS_`. The core code provides a number of key classes. Some of these classes include the following:

- Image class (i.e., `jas_image_t`). This class is used to represent an image. Methods are provided for such things as:
 - image creation/destruction,
 - querying general image properties (e.g., reference grid width and height, color profile),
 - querying component properties (e.g., width, height, grid offset, grid spacing, component type, sample precision/signedness),
 - setting various image properties,
 - loading and saving an image (i.e., encoding/decoding),
 - copying an image,
 - adding and deleting components, and
 - reading and writing component data.
- Color profile class (i.e., `jas_cmpref_t`). This class is used to define a color space. Such a definition is made relative to a reference color space such as CIE XYZ or CIE Lab [?] .
- Color transform class (i.e., `jas_cmxf_t`). This class is used to apply a color space conversion to image data. A color space transform is created from two or more color profiles.
- Stream class (i.e., `jas_stream_t`). This class provides I/O streams similar to that of standard C library [?] , but with additional functionality required by other code in the JasPer library. This extra functionality includes:
 - the ability to associate an object other than a file descriptor with a stream (such as a memory buffer), and
 - multi-character unget.
- Fixed-point number class. This templated class (i.e., a set of macros) provides a fixed-point number class. Support is provided for basic arithmetic operations, type conversion, and rounding.
- Tag-value parser class (i.e., `jas_tvp_t`). This class is used to parse strings containing one or more tag-value pairs. A tag-value pair is a string of the form "tag=value". Tag-value pairs are used by some interfaces within JasPer in order to pass parameters. For example, such pairs are used to pass options to codec drivers for encoding/decoding operations. Methods are provided for such things as:
 - creation/destruction,
 - finding the next tag-value pair in a string, and
 - querying the current tag and value.

In addition to the above classes, some other functionality is provided, including command line parsing routines (similar in spirit to UNIX `getopt`).

5.3 Codec Drivers

The core code provides a framework for housing codec drivers. A codec driver provides the means for encoding/↔ decoding of image data in a particular format. Each driver provides three methods:

1. an encoding method,
2. a decoding method, and
3. a validation method.

The encoding method emits the coded version of an image (i.e., a `jas_image_t` object) to a stream (i.e., a `jas_stream_t` object). The decoding method creates an image (i.e., a `jas_image_t` object) from the coded data in a stream. The validation method is used to quickly test if the data in a stream is formatted correctly for the image format in question. This particular method is used for the autodetection of image formats.

The codec drivers provided with the JasPer distribution are written in order to accommodate streamed data. In other words, image data streams are always processed in a single pass. This design philosophy eliminates the need for a stream object to be seekable. As a result, it is possible to write application programs that receive data from, or send data to, pipelines or other entities that do not support random access to data.

5.4 Image Model

The set of applications for which JasPer may be a useful tool is dictated, in part, by the image model that JasPer employs. Therefore, it is prudent to introduce this model here. The image model employed by JasPer is quite general and partially inspired by the one used in the JPEG-2000 standard.

An image is comprised of one or more components. In turn, each component consists of rectangular array of samples. This structure is depicted in Figures [fig_imgmodel_a](#) and [fig_imgmodel_b](#). The sample values for each component are integer valued, and can be signed or unsigned with precision from 1 to (nominally) 16 bits/sample. The maximum allowable precision is platform dependent. Most common platforms, however, should be able to accommodate at least 16 bits/sample. The signedness and precision of the sample data are specified on a per-component basis. All of the

components are associated with same spatial extent in an image, but represent different types of information.

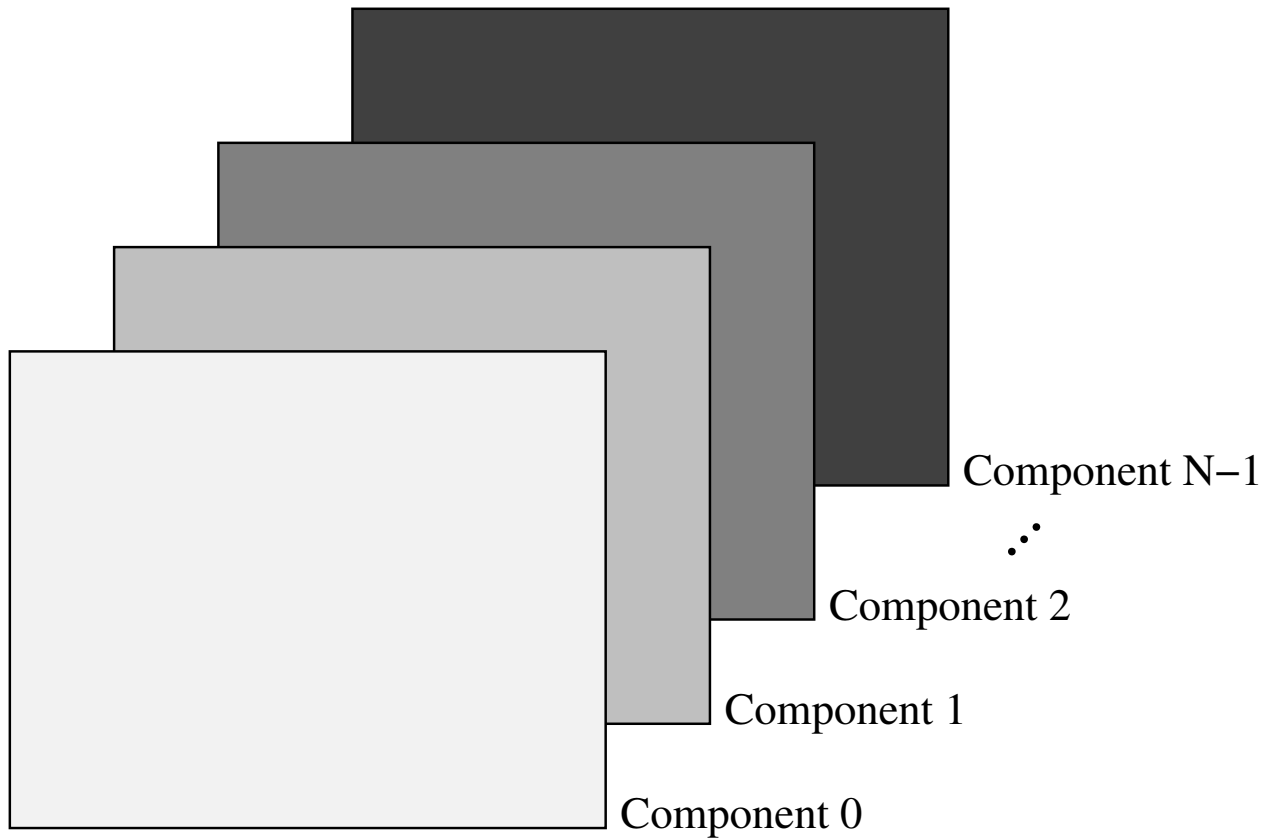


Figure 5.2 Image model: An image with N components.

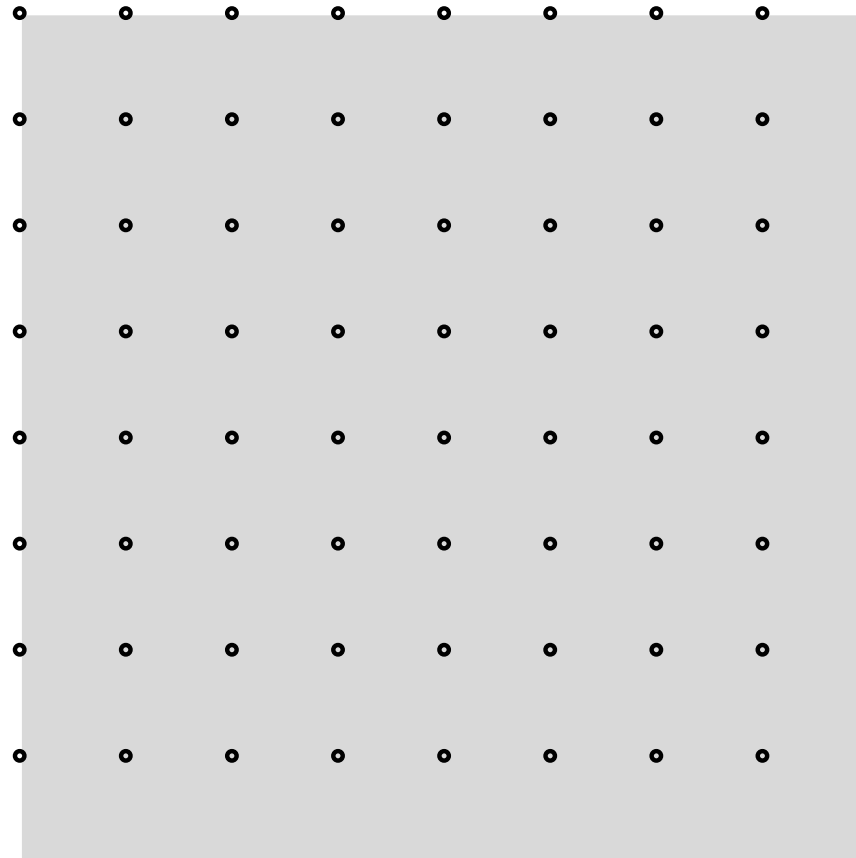


Figure 5.3 Image model: An individual component.

There is considerable flexibility in the interpretation of components. A component may represent spectral information (e.g., a color plane) or auxiliary information (e.g., an opacity plane). For example, a RGB image would have three components, where one component is associated with each of the red, green, and blue color planes. A RGBA (i.e., RGB with transparency) image would have four components, one associated with each of the red, green, blue, and alpha planes. The various components need not be sampled at the same resolution. In other words, different components may have different sampling periods. For example, when color images are represented in a luminance-chrominance color space, it is not uncommon for the luminance information to be more finely sampled than the chrominance information.

Since an image can have a number of components, a means must exist for specifying how these components are combined together in order to form a composite image. For this purpose, we employ an integer lattice known as the reference grid. The reference grid provides an anchor point for the various components of an image, and establishes their alignment relative to one another.

Each component is associated with a rectangular sampling grid. Such a grid is uniquely specified by four parameters:

1. the horizontal offset (HO),
2. vertical offset (VO),
3. horizontal spacing (HS), and
4. vertical spacing (VS).

The samples of a component are then mapped onto the points where the sampling grid intersects the reference grid. In this way, sample (i,j) of a component is mapped to the position $(\text{HO} + i \text{HS}, \text{VO} + j \text{VS})$ on the reference grid.

To clarify the above text, we now present an illustrative example. Consider an image with three components. For the k th component, let us denote the horizontal grid offset, vertical grid offset, horizontal grid spacing, and vertical grid spacing, as HO_k , VO_k , HS_k , and VS_k , respectively. Suppose, for example, that these parameters have the following values:

k	$(\text{HO}_k, \text{VO}_k)$	$(\text{HS}_k, \text{VS}_k)$
0	(0, 0)	(2, 2)
1	(2, 3)	(3, 4)
2	(3, 2)	(4, 3)

In this scenario, the component samples would be aligned on the reference grid as illustrated in Figure [fig_refgridex](#). Perhaps, it is worth noting that the above set of parameter values was chosen in order to provide an enlightening example, and is not meant to represent a set of values that is likely to be used with great frequency by applications.

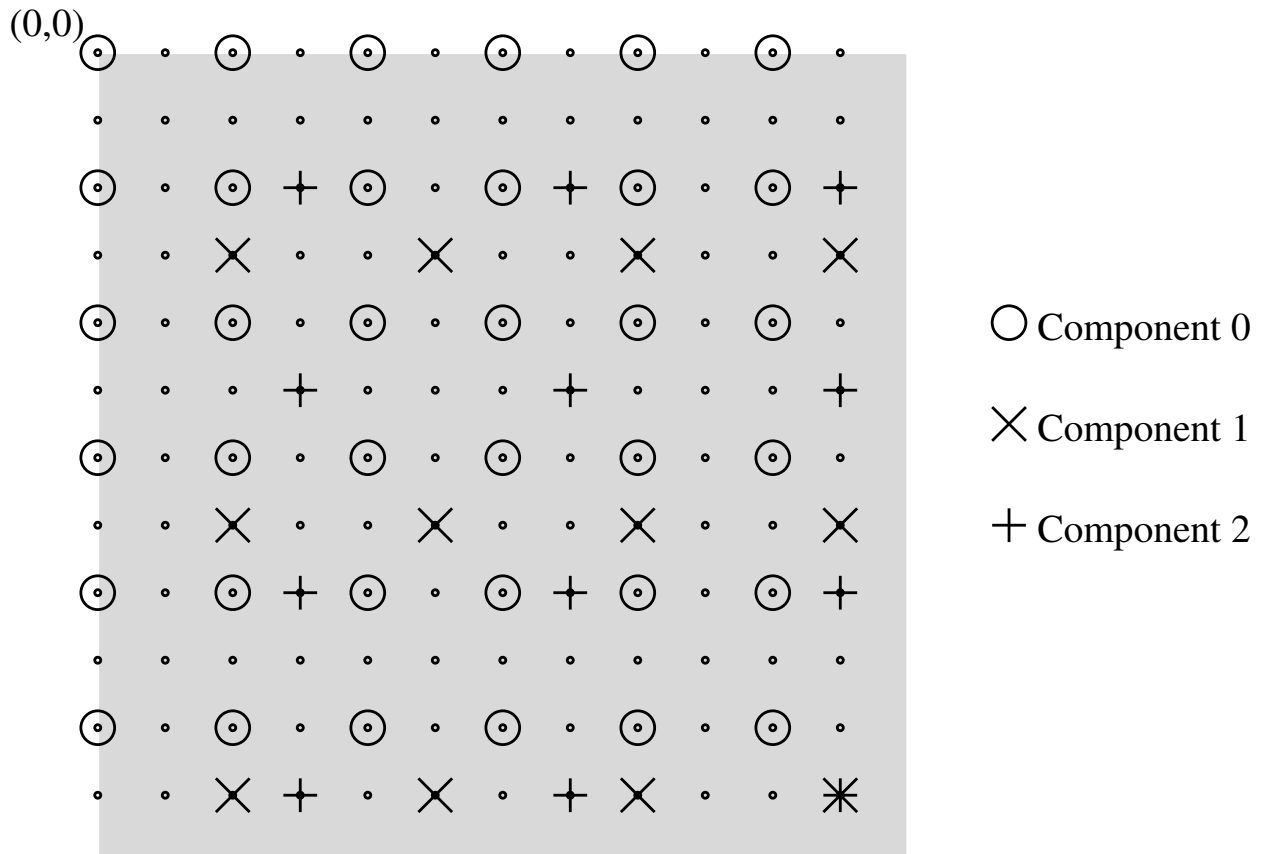


Figure 5.4 Alignment of components on the reference grid.

From above, we can see that the image model used by JasPer is quite general. The main constraint imposed by this model is that rectangular sampling must be employed. The vast majority of applications, however, use such sampling.

Also, with Jasper, one can easily accommodate grayscale, color, and other multi-band data (with or without opacity information).

Since the image model employed is true color (i.e., not palettized), the codec drivers are responsible for palettization and depalettization in the case of image formats that utilized palettized representations.

5.5 Jasper Header Files

In order to use the Jasper library, a C source file normally must include the main Jasper library header file `jasper/jasper.h`. This can be accomplished with the following preprocessor directive:

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

The main header file includes all of the other library header files. Therefore, in order to insulate application code from possible changes to the names of the other library header files, one should only ever include the main library header directly.

5.6 Initializing the Library

The first usage of the library must always be to initialize it. This is accomplished as described in [Configuration, Initialization, and Shutdown](#). If any functionality of the library is used before initialization is performed, the resulting behavior is undefined.

5.7 Memory Allocation

All memory allocation in the libjasper library is performed via the functions `jas_malloc()`, `jas_realloc()`, `jas_calloc()`, and `jas_free()`. The underlying memory allocator used by these functions can be controlled by the application. By default, an allocator based on `malloc()` is used. If one is trying to port the Jasper code to an embedded platform, it might be necessary to use a custom memory allocator instead. More detailed information on memory allocators for Jasper can be found in [Memory Allocators and the Allocator Wrapper](#).

5.8 Adding Support for a New Image Format

Support for new image formats can be easily added to Jasper. In order to add support for a new image format, one must provide three functions:

- an encoding function,
- a decoding function, and
- a validation function.

The encoding function emits the coded version of an image (i.e., an `jas_image_t` object) to a stream (i.e., a `jas_stream_t` object). The decoding function creates an image (i.e., a `jas_image_t` object) from the coded data in a stream (i.e., a `jas_stream_t` object). The validation function is used to quickly test if a data stream is formatted correctly for the image format in question. (This functionality is necessary for the autodetection of image formats.)

The precise interface provided by each of the encoding, decoding, and validation functions is as follows:

- `int (*encode)(jas_image_t *image, jas_stream_t *out, const char *opts);`
 Encode image data to a stream. The image pointed to by `image` is encoded in accordance with the options specified in the null-terminated string pointed to by `opts` and written to the stream `out`. The options string is a whitespace-delimited sequence of tag-value pairs. If the encoding operation is successful, zero is returned. Otherwise, a nonzero value is returned.
- `jas_image_t *(*decode)(jas_stream_t *in, const char *opts);`
 Decode an image from a stream. The image data from the stream `in` is decoded in accordance with the options specified in the null-terminated string pointed to by `opts`. The options string is a whitespace-delimited sequence of tag-value pairs. If the decoding operation is successful, a pointer to the decoded image is returned. Otherwise, a null pointer is returned.
- `int (*validate)(jas_stream_t *in);`
 Determine if stream data is in a particular format. The first few characters of the stream `in` are examined in order to determine if the stream is encoded in the format supported by the codec. The characters examined are not removed from the stream. (In other words, the current read/write position in the stream is left unchanged by this function.) If the format is supported by the codec, zero is returned. Otherwise, a nonzero value is returned.

Numerous examples of these types of function can be found by examining the code for the image formats already supported by JasPer (e.g., BMP, JP2, JPC, MIF, PGX, PNM, RAS, and JPEG). Once the above functions have been written, the JasPer library can be made aware of the new image format through a call to `jas_image_addfmt()`. This call, of course, must be made after the JasPer library has been initialized.

5.9 Topics

- [Configuration, Initialization, and Shutdown](#)
- [Images, Sequences, and Color Management](#)
- [I/O Streams](#)
- [Logging](#)
- [String Processing](#)
- [Math Support](#)
- [Timers](#)

5.10 Configuration, Initialization, and Shutdown

5.10.1 API References

- [Configuration, Initialization, and Shutdown](#)

5.10.2 Configuration, Initialization, and Shutdown

There are three stages in the setup of the Jasper library at run time:

1. configuration;
2. global (i.e., library-wide) initialization; and
3. per-thread initialization.

These stages of setup are performed in the order listed. Aside from the exceptions noted in this section, code in the Jasper library must not be invoked by a thread prior to per-thread initialization being performed by that thread. Only the following functions may be called prior to the configuration of the Jasper library (via `jas_conf_clear()`):

- `jas_get_total_mem_size()`

The configuration stage initializes the current (run-time) configuration settings for the library. First, the configuration settings are initialized to their default values using the `jas_conf_clear()` function (where these default values are fixed at the time that the library is built). Then, configuration settings may be changed from their defaults (prior to global initialization), if so desired. The following functions may be called after the Jasper library is configured (via `jas_conf_clear()`) but before global initialization is performed (via `jas_init_library()`):

- the `jas_conf_*` family of functions, including functions such as:
 - `jas_conf_clear()`
 - `jas_conf_set_multithread()`
 - `jas_conf_set_allocator()`
 - `jas_conf_set_debug_level()`
 - `jas_conf_set_max_mem_usage()`
 - `jas_conf_set_dec_default_max_samples()`
 - `jas_conf_set_vlogmsgf()`
- `jas_std_allocator_init()`

In order to use Jasper in multiple threads, the multithread flag must be set (to true) during configuration using the `jas_conf_set_multithread()` function.

The global initialization stage initializes the library itself using the current configuration settings. The `jas_init_library()` function performs global initialization (but not configuration). After global initialization is performed, per-thread initialization can then be performed.

The per-thread initialization stage performs any initialization necessary in order to allow a thread to use the library. The `jas_init_thread()` function performs per-thread initialization. Per-thread initialization cannot be performed until after global initialization. After a thread performs per-thread initialization, it is free to use the full API of the library. Aside from the exceptions noted in this section, no function in the library may be invoked by a thread prior to it performing per-thread initialization via `jas_init_thread()`. Per-thread initialization (via `jas_init_thread()`) can be performed at any point during the lifetime of a thread (not necessarily at the start of the thread's lifetime), subject to the constraint that the thread must not use any macros/functions in the Jasper library that require per-thread initialization to have been performed prior to calling `jas_init_thread()`.

The shutdown of the library roughly mirrors the process for library setup and consists of the following two stages:

1. per-thread clean-up
2. global (i.e., library-wide) clean-up

Each thread that has performed per-thread initialization must perform per-thread clean-up prior to global clean-up. Per-thread clean-up is performed by invoking the `jas_cleanup_thread()` function. Global clean-up is performed by invoking the `jas_cleanup_library()` function.

The library cannot be re-initialized (without first being cleaned up). That is, calling `jas_init_library()` more than once without an intervening call to `jas_cleanup_library()` is not allowed. Note, however, that a program can call `jas_init_library()` more than once, provided that the program makes an intervening call to `jas_cleanup_library()`. Similarly, a thread cannot be re-initialized (without first being cleaned up). That is, calling `jas_init_thread()` more than once without an intervening call to `jas_cleanup_thread()` is not allowed. Note, however, that a thread can call `jas_init_thread()` more than once, provided that the thread makes an intervening call to `jas_cleanup_thread()`.

Configuration, global initialization, and global clean-up should be performed on the same thread. That is, `jas_conf_clear()`, `jas_init_library()`, and `jas_cleanup_library()` must all be called from the same thread.

For backward-compatibility with older versions of the library, the `jas_init()` and `jas_cleanup()` functions may also be used for library setup and shutdown. These functions have been marked as deprecated, and will be removed in a future version of the library. Also, if these functions are used, JasPer cannot be employed in more than one thread and library configuration parameters cannot be changed from their defaults. The `jas_init()` function can be used (by legacy code) to perform configuration, global initialization, and per-thread initialization for the calling thread. The `jas_init()` function internally calls `jas_conf_clear()` to initialize the configuration settings with their default values, and then invokes `jas_init_library()` and `jas_init_thread()`. Similarly, `jas_cleanup()` can be used (by legacy code) to perform per-thread cleanup for the calling thread and global cleanup.

5.10.3 Library State

The JasPer library has the notion of a context. A context is used to store certain state used by the library. There can be multiple contexts in use at any given time. From the library user's point of view, a context is specified using an opaque handle. The type of this handle is `jas_context_t`. A handle with the special value of 0 does not correspond to a valid context. It is analogous to a null pointer.

The JasPer library allows the current context to be set independently for each thread. That is, the current context is a per-thread setting. When a thread is initialized, a single context is automatically created, which is referred to as the default context, and the current context for the thread is set to this default context. A context cannot be shared by more than one thread, as this would lead to data races and other synchronization problems.

The JasPer library has two types of state:

1. Global state. That is, state that is library wide.
2. Per-context state. That is, state that only applies to a particular context.

Most state maintained by the library is in the form of per-context state. This is motivated by the desire to minimize the sharing of state between threads, which would require locking/synchronization. The global state for the library includes:

- the memory allocator to be used by the library
- the maximum amount of memory that the library is permitted to use
- certain configuration settings that are used as default values when creating/initializing contexts

The per-context state includes:

- the debug level
- the `vlogmsgf` function to be used for logging error/warning/informational messages
- the image format table
- the maximum number of samples in an image that a decoder is allowed to process

5.10.4 Contexts

Contexts can be created and destroyed with the following functions:

- `jas_context_create()`
- `jas_context_destroy()`

The current context for the calling thread can be set/queried with the following functions:

- `jas_get_context()`
- `jas_set_context()`

The various per-context settings can be set/queried with the following functions:

- `jas_set_debug_level()`
- `jas_get_debug_level()`
- `jas_set_dec_default_max_samples()`
- `jas_get_dec_default_max_samples()`
- `jas_set_vlogmsgf()`
- `jas_get_vlogmsgf()`

5.10.5 Example of Code for Library Setup and Shutdown

Setup and shutdown of the library would typically be performed in the main thread of the application using code resembling the following:

```
/*
Configure the library using the default configuration settings.
*/
jas_conf_clear();

/*
Change any configuration parameters for which defaults are not suitable
by using the jas_conf_* family of functions.
An example of this type of code follows.
*/

static jas_std_allocator_t allocator;
jas_std_allocator_init(&allocator);
jas_conf_set_allocator(JAS_CAST(std_allocator_t *, &allocator));
jas_conf_set_max_mem_usage(10000000);
jas_conf_set_multithread(true);

/*
Perform global initialization for the Jasper library.
*/
if (jas_init_library()) {
    /* Handle the initialization error. */
}

/*
Perform any per-thread initialization for the Jasper library.
*/
if (jas_init_thread()) {
    /* Handle the initialization error. */
}

/*
Use the Jasper library.
*/

/* Perform any per-thread clean-up for the Jasper library. */
jas_cleanup_thread();

/* Perform global clean-up for the Jasper library. */
jas_cleanup_library();
```

In the case of an application where the Jasper library is used in more than one thread, each additional thread would need to perform per-thread initialization (via `jas_init_thread()`) before using the library. Also, each additional thread would need to perform per-thread cleanup (via `jas_cleanup_thread()`) when the use of the library is no longer needed. For each thread, the code for this might resemble something like the following:

```
/*
Perform any per-thread initialization for the Jasper library.
*/
if (jas_init_thread()) {
    /* Handle the initialization error. */
}

/*
Use the Jasper library.
*/

/* Perform any per-thread clean-up for the Jasper library. */
jas_cleanup_thread();
```

When practical, it is probably preferable (for reasons of efficiency) to invoke `jas_init_thread()` and `jas_cleanup_thread()` only once in the lifetime of a thread. In some cases, this may be difficult to do, due to the structure of the application code. In such situations, one could simply wrap code using the Jasper library in calls to `jas_init_thread()` and `jas_cleanup_thread()`. As explained earlier, this is valid to do. For example, code like that shown in the example below is valid (assuming that the global initialization of the Jasper library has already been performed).

```
/* Code that does not use the Jasper library. */

/* Perform per-thread initialization of the Jasper library. */
```



```

if (jas_init_thread()) {
    /* Handle initialization error. */
}

/* Code that uses the Jasper library (e.g., performing encoding/decoding).

/* Perform per-thread cleanup of the Jasper library. */
jas_cleanup_thread();

/* Code that does not use the Jasper library. */

/* Perform per-thread initialization of the Jasper library. */
if (jas_init_thread()) {
    /* Handle initialization error. */
}

/* Code that uses the Jasper library (e.g., performing encoding/decoding).

/* Perform per-thread cleanup of the Jasper library. */
jas_cleanup_thread();

/* Code that does not use the Jasper library. */

```

5.10.6 Additional Examples of Library Setup

Some additional examples of using the `jas_conf_clear()`, `jas_init_library()`, `jas_init_thread()`, `jas_cleanup_thread()`, and `jas_cleanup_library()`, functions (as well as the `jas_init()` and `jas_cleanup()` functions) can be found in the source code for the application programs `jasper`, `imginfo`, and `imgcmp`. Moreover, the application program `multithread` is an example of a program that uses the Jasper library in multiple threads.

5.10.7 Memory Allocators and the Allocator Wrapper

The library provides a simple interface for memory allocators. This is provided through the `jas_allocator_t` type. An allocator object resides in memory managed by the library user. If the library user invokes `jas_cleanup_library()` via `atexit()`, then the allocator should obviously not be allocated on the stack. (The `jas_cleanup_library()` function will use the allocator in order to free memory previously allocated by the library.)

The `jas_std_allocator_init()` function provides a way to create an allocator that uses `malloc()` and related functions from the C standard library.

The Jasper library does not directly use the allocator provided by the library user. It instead uses this allocator indirectly through a wrapper. The allocator wrapper is a pseudo-allocator. That is, it does not actually allocate memory directly but rather delegates the memory allocation operations to another allocator (namely, the one specified by the library user). The allocator wrapper tracks the amount of memory used by the allocator to which it delegates. This eliminates the need for the library user's allocator to track this information itself.

As long as the allocator wrapper functionality is enabled, the Jasper library will track how much memory is being used by the allocator in order to allow a limit to be imposed on memory usage. The allocator wrapper composes with the allocator selected by the library user, as explained above. So, this memory limiting functionality is available even when the library user provides a custom allocator that does not itself track memory usage.

Although `jas_malloc()`, `jas_realloc()`, `jas_free()` and other related functions internally use the allocator provided by the library user for all memory allocations, this does not imply that pointers returned by `jas_malloc()` (and related functions) can be used with the library user's allocator directly. For this reason, it is important to use the `jas_free()` function to free memory allocated by `jas_malloc()` and friends (and not attempt to directly invoke the underlying custom allocator provided by the library user).

An allocator provides functions with the following signatures and semantics:

- `void (*cleanup)(jas_allocator_t *allocator);`

This function performs a clean-up operation, which cleans up the allocator when it is no longer needed. This operation should free any resources associated with the allocator. The allocator cannot be used after the clean-up operation is performed. This function pointer may be null, in which case the clean-up operation is treated as a no-op.

- `void *(*alloc)(jas_allocator_t *allocator, size_t size);`

This function performs a memory-allocation operation, and has behavior similar to `malloc()`.

- `void (*free)(jas_allocator_t *allocator, void *pointer);`

This function performs a memory-deallocation operation, and has behavior similar to `free()`.

- `void *(*realloc)(jas_allocator_t *allocator, void *pointer, size_t new_size);`

This function performs a memory-reallocation operation, and has behavior similar to `realloc()`.

5.10.8 Logging

All error, warning, informational, and debugging messages are normally generated via the logging interface provided by the library. The library user can specify a special function used to generate formatted logging messages. The library user can therefore control where messages are directed. The function has the following signature:

- `int (*vlogprintf)(jas_logtype_t type, const char *format, va_list ap);`

The function formats and outputs a log message. The interface of this function is somewhat similar to `vprintf`. In addition to a format string and items used for formatting, the function also takes a specification of the type of message being generated (e.g., error, warning, etc.).

This function is used in order to implement functions such as:

- `jas_vlogmsgf()`
- `jas_logprintf()`
- `jas_logerrorf()`
- `jas_logwarnf()`
- `jas_loginfof()`
- `jas_logdebugf()`

5.11 Images, Sequences, and Color Management

5.11.1 API References

- [Image Representation](#)
- [Color Management](#)
- [One- and Two-Dimensional Sequences](#)

5.11.2 Images

An image is represented by the type `jas_image_t`. An image component is represented by the type `jas_image_cmpt_t`.

The image-format table contains an entry for each codec that is supported by the Jasper library. More specifically, an entry in the table will be present for each codec whose support was included in the Jasper library when the library was built. A codec may be either in an enabled or disabled state. The default value for the enable/disable setting is specified when the library is built. The enable/disable setting can also be changed (from the default) at run time. Most functions that query the image-format table ignore (or skip) entries that correspond to disabled codecs. The default build-time configuration settings for Jasper are such that codecs that are experimental or dangerous (due to posing security risks) are disabled by default.

The following functions are provided for image creation/destruction:

- `jas_image_create()`
- `jas_image_create0()`
- `jas_image_copy()`
- `jas_image_destroy()`

The following functions are provided for image encoding/decoding:

- `jas_image_decode()`
- `jas_image_encode()`

The following functions are provided for various other functionality for images:

- `jas_image_rawsize()`
- `jas_image_ishomosamp()`
- `jas_image_cmprof()`
- `jas_image_sampcmpt()`
- `jas_image_writecmpt2()`
- `jas_image_readcmpt2()`
- `jas_image_chclrspc()`
- `jas_image_dump()`

The following functions are provided for accessing/manipulating image components:

- `jas_image_cmptwidth()`
- `jas_image_cmptheight()`

- `jas_image_cmptsgnd()`
- `jas_image_cmptprec()`
- `jas_image_cmptthstep()`
- `jas_image_cmptvstep()`
- `jas_image_cmpttlx()`
- `jas_image_cmpttly()`
- `jas_image_cmptbrx()`
- `jas_image_cmptbry()`
- `jas_image_cmpt_domains_same()`
- `jas_image_readcmpt()`
- `jas_image_writecmpt()`
- `jas_image_delcmpt()`
- `jas_image_addcmpt()`
- `jas_image_copycmpt()`
- `JAS_IMAGE_CDT_GETSGND()`
- `JAS_IMAGE_CDT_SETSGND()`
- `JAS_IMAGE_CDT_GETPREC()`
- `JAS_IMAGE_CDT_SETPREC()`
- `jas_image_cmptdtype()`
- `jas_image_depalettize()`
- `jas_image_readcmptsample()`
- `jas_image_writecmptsample()`
- `jas_image_getcmptbytype()`

The following functions are provided for accessing/manipulating the image-format table:

- `jas_image_clearfmts()`
- `jas_image_addfmt()`
- `jas_image_lookupfmtbyid()`
- `jas_image_lookupfmtbyname()`
- `jas_image_fmtfromname()`
- `jas_image_getfmt()`
- `jas_image_strtofmt()`
- `jas_image_fmtostr()`
- `jas_image_getnumfmts()`
- `jas_image_getfmtbyind()`

5.11.3 Color Management

The Jasper library provides basic support for color management. A color-management (CM) profile is used to describe the color space used by an image. The following function are provided for accessing/manipulating CM profiles:

- `jas_cmxform_create()`
- `jas_cmxform_destroy()`
- `jas_cmxform_apply()`
- `jas_cmprof_createfromclrspc()`
- `jas_cmprof_destroy()`
- `jas_clrspc_numchans()`
- `jas_cmprof_clrspc()`
- `jas_cmprof_copy()`

The following functions are provided for accessing/manipulating color spaces:

- `jas_clrspc_create()`
- `jas_clrspc_fam()`
- `jas_clrspc_mbr()`
- `jas_clrspc_isgeneric()`
- `jas_clrspc_isunknown()`

5.11.4 ICC Profiles

The Jasper library offers support for managing ICC profiles. The following functions are provided for managing such profiles:

- `jas_iccprof_load()`
- `jas_iccprof_save()`
- `jas_iccprof_destroy()`
- `jas_iccprof_getattr()`
- `jas_iccprof_setattr()`
- `jas_iccprof_dump()`
- `jas_iccprof_copy()`
- `jas_iccprof_gethdr()`
- `jas_iccprof_sethdr()`

- `jas_iccattrval_destroy()`
- `jas_iccattrval_dump()`
- `jas_iccattrval_allowmodify()`
- `jas_iccattrval_clone()`
- `jas_iccattrval_create()`
- `jas_iccattrtab_dump()`
- `jas_iccprof_createfrombuf()`
- `jas_iccprof_createfromclrspc()`

5.11.5 Sequences and Matrices

The JasPer library provides classes for representing matrices and both 1-D and 2-D sequences.

Matrices are represented in JasPer using the `jas_matrix_t` class. A list of the various functions/macros provided for this class is follows:

- matrix creation/destruction:
 - `jas_matrix_create()`
 - `jas_matrix_copy()`
 - `jas_matrix_destroy()`
- getting/setting various attributes of a matrix:
 - `jas_matrix_numrows()`
 - `jas_matrix_numcols()`
 - `jas_matrix_size()`
 - `jas_matrix_empty()`
 - `jas_matrix_resize()`
 - `jas_matrix_rowstep()`
 - `jas_matrix_step()`
- getting/setting elements of a matrix:
 - `jas_matrix_get()`
 - `jas_matrix_set()`
 - `jas_matrix_getv()`
 - `jas_matrix_setv()`
 - `jas_matrix_getref()`
 - `jas_matrix_getvref()`
- accessing submatrices of a matrix:
 - `jas_matrix_bindsub()`
 - `jas_matrix_bindrow()`

- `jas_matrix_bindcol()`
- performing an arithmetic operations on all elements of a matrix:
 - `jas_matrix_clip()`
 - `jas_matrix_asl()`
 - `jas_matrix_asr()`
 - `jas_matrix_divpow2()`
 - `jas_matrix_setall()`
- relational operations for matrices:
 - `jas_matrix_cmp()`
- reading and writing a matrix from and to a stream:
 - `jas_matrix_input()`
 - `jas_matrix_output()`

Sequences in 2-D are represented using the `jas_seq2d_t` class. In effect, a 2-D sequence is simply a matrix whose starting row and column indices are arbitrary (instead of being fixed at zero as in the case of a matrix). A list of the various functions/macros provided for this class is follows:

- creation/destruction of a sequence:
 - `jas_seq2d_create()`
 - `jas_seq2d_destroy()`
 - `jas_seq2d_copy()`
- getting/setting attributes of a sequence:
 - `jas_seq2d_xstart()`
 - `jas_seq2d_ystart()`
 - `jas_seq2d_xend()`
 - `jas_seq2d_yend()`
 - `jas_seq2d_rowstep()`
 - `jas_seq2d_width()`
 - `jas_seq2d_height()`
 - `jas_seq2d_setshift()`
 - `jas_seq2d_size()`
 - `jas_seq2d_empty()`
- accessing elements of a sequence:
 - `jas_seq2d_getref()`
 - `jas_seq2d_get()`
- referencing a subsequence of a sequence:
 - `jas_seq2d_bindsub()`

Sequences in 1-D are represented using the `jas_seq_t` class. A list of the various functions/macros provided for this class is follows:

- creation/destruction of a sequence:
 - `jas_seq_create()`
 - `jas_seq_destroy()`
- querying the attributes of a sequence:
 - `jas_seq_start()`
 - `jas_seq_end()`
- accessing the elements of a sequence:
 - `jas_seq_set()`
 - `jas_seq_getref()`
 - `jas_seq_get()`

5.12 I/O Streams

5.12.1 API References

- [I/O Streams](#)

5.12.2 I/O Streams

The library provides a type for representing a stream of characters for input/output. Such a stream is represented by the type `jas_stream_t`. Streams (in the JasPer library) are similar to I/O streams in the C standard library but have a few additional functionalities. In particular, with JasPer I/O streams, it is possible to:

- set a limit on the number of characters that can be read from or written to a stream;
- put back more than one character read from a stream;
- query the number of characters that have been read from or written to a stream; and
- associate a stream with one of several different types of underlying data sources/sinks (including buffers in memory and temporary files).

The possible underlying sources/sinks for a stream include:

- a stdio stream (i.e., `FILE`)
- a file descriptor
- a memory buffer
- a temporary file

The following functions are used for opening/closing streams:

- `jas_stream_fopen()`
- `jas_stream_memopen2()`
- `jas_stream_fdopen()`
- `jas_stream_freopen()`
- `jas_stream_tmpfile()`
- `jas_stream_close()`

The following functions are used for reading/writing streams:

- `jas_stream_read()`
- `jas_stream_write()`
- `jas_stream_peek()`
- `jas_stream_printf()`
- `jas_stream_puts()`
- `jas_stream_gets()`
- `jas_stream_ungetc()`
- `jas_stream_getc()`
- `jas_stream_putc()`
- `jas_stream_peekc()`
- `jas_stream_gobble()`
- `jas_stream_pad()`

The following functions are used for getting/setting the position within a stream:

- `jas_stream_isseekable()`
- `jas_stream_seek()`
- `jas_stream_tell()`
- `jas_stream_rewind()`

Numerous other functions for streams are also provided, including:

- `jas_stream_flush()`
- `jas_stream_copy()`
- `jas_stream_display()`

- `jas_stream_length()`
- `jas_stream_eof()`
- `jas_stream_error()`
- `jas_stream_clearerr()`
- `jas_stream_setrwlimit()`
- `jas_stream_setrwcountrwcount()`
- `jas_stream_getrwlimit()`
- `jas_stream_getrwcountrwcount()`

5.13 Logging

5.13.1 API References

- [Logging](#)

5.13.2 Logging

The library provides an interface for generating log messages. Such messages are used to convey errors, warnings, and other information. Each log message has a type. The type of the log message is represented by the type `jas_logtype_t`. This type has two components:

- a class (e.g., error, warning, informational, debugging)
- a priority (which is typically only used for debugging messages)

The class and priority of a log message can be obtained from its type by using the functions:

- `jas_logtype_getclass()`
- `jas_logtype_getpriority()`

A log type can be initialized via the function:

- `jas_logtype_init()`

Most error/warning and other messages generated by the Jasper library (including its codecs) use the logging interface. In particular, the following functions are used for logging:

- `jas_vlogmsgf()`
- `jas_logprintf()`
- `jas_logerrorf()`
- `jas_logwarnf()`
- `jas_loginfof()`
- `jas_logdebugf()`

All of these functions are ultimately routed through the `vlogmsgf` function provided by the library user. The default behavior of this function is to write messages to the standard error stream.

5.14 String Processing

5.14.1 API References

- [Command-Line Interface \(CLI\) Option Processing](#)
- [Tag-Value Pair \(TVP\) Parsing](#)
- [String Processing](#)

5.14.2 Command-Line Interface (CLI) Option Processing

The JasPer library provides the following function for processing command-line interface (CLI) options:

- [jas_getopt\(\)](#) This function is somewhat similar the `getopt()` function in the POSIX standard but provides support for both long and short option names.

Each CLI option is described using an object of the type [jas_opt_t](#). The set of full CLI options is specified as an array of such objects.

5.14.3 Tag-Value Pair (TVP) Processing

A tag-value pair (TVP) is a string of the form "tag=value".

The JasPer library provides functionality for parsing tag-value pairs in strings. This functionality is provided through the [jas_tvparser_t](#) class.

- [jas_tvparser_create\(\)](#)
- [jas_tvparser_destroy\(\)](#)
- [jas_taginfos_lookup\(\)](#)
- [jas_taginfo_nonull\(\)](#)
- [jas_tvparser_next\(\)](#)
- [jas_tvparser_gettag\(\)](#)
- [jas_tvparser_getval\(\)](#)

When parsing TVPs, each tag is associated with a unique integer ID. The TVP parser represents a parsed TVP using the [jas_taginfo_t](#) class, which contains a tag ID and value string.

5.14.4 String Processing

The JasPer library provides a few functions for performing basic string processing:

- `jas_strdup()`. This function is similar to the popular `strdup()` function, except that memory is allocated with `jas_malloc()` instead of `malloc()`.
- `jas_strtok()` This function is a re-entrant (i.e., thread-safe) version of the `strtok()` function in the C standard library, and is similar to the `strtok_r()` function in the POSIX standard.
- `jas_stringtokenize()` This function can be used to split a string into tokens based on a specified set of delimiters.

5.15 Math Support

5.15.1 API References

- [Fixed-Point Arithmetic](#)

5.15.2 Math Support

The JasPer library provides various functions/macros for performing fixed-point arithmetic. It also provides some functions for performing safe integer arithmetic (i.e., integer arithmetic with overflow checking). The details of these macros/functions can be found using the link in the API References section above.

5.16 Timers

5.16.1 API References

- [Timers](#)

5.16.2 Timers

The JasPer library provides a simple timer class, which can be used more measuring elapsed time.

A timer is represented using the `jas_tmr_t` class. A list of the various functions/macros provided for this class is given below.

- starting and stopping a timer:
 - `jas_tmr_start()`
 - `jas_tmr_stop()`
- querying the elapsed time for a timer:
 - `jas_tmr_get()`

Chapter 6

JasPer Application Programs

6.1 Introduction

In order to demonstrate how the JasPer library can be used, several sample application programs are provided in the JasPer software distribution. These programs include the following:

- `jasper`. The `jasper` program is an image transcoder (i.e., it converts image data from one format to another). For more details, see [The jasper Program](#).
- `jiv`. The `jiv` program is a simple image viewer (based on OpenGL). For more details, see [The jiv Program](#).
- `imgcmp`. The `imgcmp` program is an image comparison utility. It measures the difference between two images using one of numerous distortion metrics (such as peak signal-to-noise ratio, mean squared error, root mean squared error, peak absolute error, and mean absolute error). For more details, see [The imgcmp Program](#).
- `imginfo`. The `imginfo` program provides basic information about an image, such as its geometry (i.e., number of components, width and height of components, and so on). For more details, see [The imginfo Program](#).

Although the above-mentioned programs were initially developed for demonstration purposes, they have also proven quite useful in their own right, especially the `jasper` and `jiv` programs.

6.2 The jasper Program

6.2.1 The jasper Program

6.2.1.1 Synopsis

```
jasper [options]
```

6.2.1.2 Description

The jasper command converts image data from one format to another. In other words, this command functions as a general-purpose transcoder. Since the JPEG-2000 format is supported by this software, it can be used as a JPEG-2000 encoder and/or decoder.

6.2.1.3 Options

The jasper program accepts the following options:

- `--help`
 - Print usage information and exit.
- `--version.`
 - Print the version and exit.
- `--verbose`
 - Enable verbose mode.
- `--list-enabled-formats`
 - Print the names of all of the enabled image formats to standard output.
- `--memory-limit $n`
 - Set the memory limit to `$n` bytes.
- `--debug-level $level`
 - Set the debug level to `$level`.
- `--input $file` or `-f $file`
 - Read the input image from the file named `$file`. By default, the input image is read from standard input.
- `--input-format $format` or `-t $format`
 - Specify the format of the input image as `$format`. In most circumstances, this option should not be needed, as the format is normally autodetected by examining the image data directly or deduced from the input file name extension if an input file is specified (via the `--input` option).
- `--input-option $option` or `-o $option`
 - Provide the option `$option` to the decoder. The valid values for the argument `$option` are determined by the input image format. See below for more details.
- `--output $file` or `-F $file`
 - Write the encoded image to the file named `$file`. By default, the encoded image is written to standard output.
- `--output-format $format` or `-T $format`
 - Produce the output image in the format indicated by `$format`. The output format must be specified if an output file is not given (via the `--output` option). If an output file is given and no output format is specified, an attempt will be made to deduce the correct format from the output file name extension.

- `--output-option $option` or `-O $option`
 - Provide the option `$option` to the encoder. The valid values for the argument `$option` are determined by the output format. See below for more details.
- `--force-srgb`
 - Force the image to be converted to the sRGB color space before encoding. As a side effect, the image will also be homogeneously sampled (i.e., all components are sampled at the same points on the reference grid).

Assuming that JasPer is built with all codec support included (so that all codecs are available for use), the argument `$format` must have one of the following values:

- `bmp`: Windows BMP
- `heic`: HEIC Format
- `jp2`: JPEG-2000 JP2
- `jpc`: JPEG-2000 Code Stream
- `jpg`: JPEG
- `pgx`: PGX
- `pnm`: PNM/PGM/PPM
- `mif`: My Image Format
- `ras`: Sun Rasterfile

A list of the available formats is included in the help information for the program (obtained via the `--help` option).

6.2.1.4 Examples

To obtain information on how to use the `jasper` program, use the command:

```
jasper --help
```

Suppose that we have an image stored in the PNM/PPM format in a file called `lena.ppm`. To encode this image (losslessly) in the JPEG-2000 JP2 format, and store the result in a file called `lena.jp2`, use the command:

```
jasper --input lena.ppm --output lena.jp2 --output-format jp2
```

Or, alternately (using short option names), use the command:

```
jasper -f lena.ppm -F lena.jp2 -T jp2
```

Suppose that we have a RGB color image stored in the JPEG-2000 JP2 format in a file called `lena.jp2`. To encode this image in the PNM/PPM format, and store the result in a file called `lena.ppm`, use the command:

```
jasper --input lena.jp2 --output lena.ppm --output-format pnm
```

Or, alternately (using short option names), use the command:

```
jasper -f lena.jp2 -F lena.ppm -T pnm
```

Suppose that we have an image stored in the BMP format in a file called `lena.bmp`. To encode this image in a lossy manner at 100:1 compression in the JPEG-2000 (code stream) format, and store the result in a file called `lena_lossy.jpc`, use the command:

```
jasper -f lena.bmp -F lena_lossy.jpc -T jpc -O rate=0.01
```

Suppose that we have an image stored in a file called `sachie.pnm` in the PNM/PPM format, and we want to encode the image in the JPEG-2000 (code stream) format and store the result in a file named `sachie_new.jpc`. Further, suppose that we want the JPEG-2000 format to employ the following parameters:

- code blocks are 64 samples in width and 32 samples in height
- no multicomponent transform is to be employed
- 4 resolution levels should be employed for each component
- the compression is lossy at 64:1

In order to accomplish the above, type:

```
jasper -f sachie.pnm -F sachie_new.jpc -T jpc -O cblkwidth=64 \
-O cblkheight=32 -O nomct -O numrlvls=4 -O rate=0.015625
```

6.3 The jiv Program

6.3.1 The jiv Program

6.3.1.1 Synopsis

```
jiv [options] [file1 file2 ...]
```

6.3.1.2 Description

The `jiv` command displays an image. Basic pan and zoom functionality is provided. Components of an image may be viewed individually. Color components may also be viewed together as a composite image. At present, the `jiv` image viewer has only trivial support for color. It recognizes RGB and YCbCr color spaces, but does not use tone reproduction curves and the like in order to accurately reproduce color. For basic testing purposes, however, the color reproduction should suffice.

6.3.1.3 Options

The following options are supported:

- `--help`
 - Print help information and exit.
- `--version`
 - Display the version information and exit.
- `--wait $n`
 - Automatically step from one image to the next, pausing for `$n` seconds in between.
- `--loop`
 - Repeatedly cycle through the specified images.
- `--memory-limit $n`
 - Set the memory limit to `$n` bytes.
- `--debug-level $level`
 - Set the debug level to `$level`.

6.4 The imgcmp Program

6.4.1 The imgcmp Program

6.4.1.1 Synopsis

```
imgcmp [options]
```

6.4.1.2 Description

The `imgcmp` command compares two images. The two images being compared must have the same geometry (i.e., the same width, height, number of components, component subsampling factors, etc.).

6.4.1.3 Options

The following options are supported:

- `--help`
 - Print help information and exit.
- `--version`
 - Display the version information and exit.
- `--verbose`
 - Increase the verbosity level.
- `-f $file`
 - Read the primary (i.e., reference) image (for comparison purposes) from the file named `$file`.
- `-F $file`
 - Read the secondary image (for comparison purposes) from the file named `$file`.
- `-m $metric`
 - Use the difference metric specified by `$metric`. The `$metric` argument may assume one of the following values:
 - * `psnr`: peak signal to noise ratio (PSNR)
 - * `mse`: mean squared error (MSE)
 - * `rmse`: root mean squared error (RMSE)
 - * `pae`: peak absolute error (PAE)
 - * `mae`: mean absolute error (MAE)
 - * `equal`: equality
- `--memory-limit $n`
 - Set the memory limit to `$n` bytes.
- `--debug-level $level`
 - Set the debug level to `$level`.

The `-f` and `-F` options must always be specified. There is currently no way to explicitly specify the format of the images. If the format of either image cannot be autodetected, the command will exit with an error.

6.4.1.4 Examples

Suppose that we have two slightly different versions of an image stored in files `original.pgm` and `reconstructed.pgm`. In order to calculate the difference between these images using the PSNR metric, use the command:

```
imgcmp -f original.pgm -F reconstructed.pgm -m psnr
```

6.5 The imginfo Program

6.5.1 The imginfo Program

6.5.1.1 Synopsis

`imginfo [options]`

6.5.1.2 Description

The `imginfo` command displays information about an image. This command is really only intended to be used from shell scripts for testing purposes.

6.5.1.3 Options

The following options are supported:

- `--help`
 - Print help information and exit.
- `--version`
 - Display the version information and exit.
- `--verbose`
 - Increase the verbosity level.
- `-f $file`
 - Specify that the image should be read from the file named `$file`. If this option is not specified, the image is read from standard input.
- `--list-enabled-formats`
 - Print the names of all of the enabled image formats to standard output.
- `--memory-limit $n`
 - Set the memory limit to `$n` bytes.
- `--debug-level $level`
 - Set the debug level to `$level`.
- `--max-samples $n`
 - Set the maximum number of samples for decoding to `$n`.
- `--decoder-option $string`
 - Add the option `$string` to the list of decoder options.

6.5.1.4 Examples

To obtain information on how to use the `imginfo` program, use the command:

```
imginfo --help
```

Suppose that we would like to print information about an image stored in the file name `image.jp2`. This could be accomplished by using the command:

```
imginfo < image.jp2
```

Alternatively, the following command could be used:

```
imginfo -f image.jp2
```

Chapter 7

Codecs

The JasPer software supports several popular codecs. In some cases, the support is native (i.e., built-in to the JasPer library itself). In other cases, the support is provided indirectly through external software dependencies. The sections that follow describe the various codecs supported by JasPer in more detail.

7.1 Common Codec Functionality

Some encoder and decoder options are generic in the sense that they are essentially independent of the codec. Typically, encoders and decoders tend to ignore unrecognized options (although a warning might be issued for such options). So, it is not usually problematic to specify an unsupported option to an encoder or a decoder.

7.1.1 Generic Encoder Options

The following generic options are supported by some encoders:

- `debug=$level`. Set the debug level to `$level`.

7.1.2 Generic Decoder Options

The following generic options are supported by some decoders:

- `debug=$level`. Set the debug level to `$level`.
- `max_samples=$n`. Set the maximum number of samples that are permitted in an image to be decoded to `$n`. (An 100×100 RGB-color image has 30000 samples.)

7.2 BMP Codec

One of the most popular image formats on the Microsoft Windows platform is Microsoft's BMP format. The BMP codec in Jasper was written without the benefit of the BMP format specification from Microsoft. This means that the BMP support will inevitably not work correctly for all valid BMP files.

7.2.1 Encoder Options

The BMP encoder does not support any special options.

7.2.2 Decoder Options

The BMP decoder supports the following generic options:

- `max_samples`

The BMP decoder does not support any options beyond generic ones.

7.3 HEIC Codec

The HEIC format is quite popular on Apple platforms. The support is experimental and likely has numerous bugs.

7.3.1 Encoder Options

The HEIC encoder supports the following special options:

- `version`. Print the version of the Libheif library being used without performing any decoding.
- `quality $quality`. Set the quality factor to `$quality`.

7.3.2 Decoder Options

The HEIC decoder supports the following special options:

- `version`. Print the version of the Libheif library being used without performing any decoding.

The HEIC decoder does not support any generic options.

7.4 JP2 Codec

One of the two image formats specified in the JPEG-2000 Part-1 standard (i.e., ISO/IEC 15444-1 [?]) is the so called "JP2" format.

7.4.1 Encoder Options

The JP2 encoder supports all of the same options as the JPC encoder.

7.4.2 Decoder Options

The JP2 decoder supports all of the same options as the JPC decoder.

7.5 JPC Codec

One of the two image formats specified in the JPEG-2000 Part-1 standard (i.e., ISO/IEC 15444-1 [?]) is the so called JPEG-2000 code stream format. The JPC codec in JasPer implements this format.

The design of the JPEG-2000 codec implementation was driven by several key concerns: execution speed, memory usage, robustness, portability, modularity, maintainability, and extensibility. In some cases, however, during the design process, modularity, portability, and understandability of the code were weighed more heavily than execution speed and memory usage. Code understandability and portability were critical considerations since this software was intended to be used as a reference implementation of the JPEG-2000 Part-1 codec in the JPEG-2000 Part-5 standard [?] .

Since the JPEG-2000 standard does not specify any means for encoding color space information in a JPEG-2000 code stream, the decoder must make certain assumptions about the color space of an image. If accurate color representation is important, the JPEG-2000 code stream format should not be employed. The JPEG-2000 JP2 format should be used instead.

7.5.1 Encoder Options

The JPC encoder supports the following special options:

- `debug=$level`. Set the debug level to `$level`.
- `imgareatlx=$x`. Set the x-coordinate of the top-left corner of the image area to `$x`.
- `imgareatly=$y`. Set the y-coordinate of the top-left corner of the image area to `$y`.
- `tilegrdtlx=$x`. Set the x-coordinate of the top-left corner of the tiling grid to `$x`.
- `tilegrdtly=$y`. Set the y-coordinate of the top-left corner of the tiling grid to `$y`.
- `tilewidth=$w`. Set the nominal tile width to `$w`.
- `tileheight=$h`. Set the nominal tile height to `$h`.

- `prcwidth=$w`. Set the precinct width to `$w`. The argument `$w` must be an integer power of two. The default value is 32768.
- `prcheight=$h`. Set the precinct height to `$h`. The argument `$h` must be an integer power of two. The default value is 32768.
- `cblkwidth=$w`. Set the nominal code block width to `$w`. The argument `$w` must be an integer power of two. The default value is 64.
- `cblkheight=$h`. Set the nominal code block height to `$h`. The argument `$h` must be an integer power of two. The default value is 64.
- `mode=$m`. Set the coding mode to `$m`. The argument `$m` must have one of the following values:
 - `int`. integer mode
 - `real`. real mode

If lossless coding is desired, the integer mode must be used. By default, the integer mode is employed. The choice of mode also determines which multicomponent and wavelet transforms (if any) are employed.

- `rate=$r`. Specify the target rate. The argument `$r` is a positive real number. Since a rate of one corresponds to no compression, one should never need to explicitly specify a rate greater than one. By default, the target rate is considered to be infinite.
- `ilyrrates= r_0, r_1, \dots, r_{N-1}` . Specify the rates for any intermediate layers. The argument to this option is a comma separated list of N rates. Each rate is a positive real number. The rates must increase monotonically. The last rate in the list should be less than or equal to the overall rate (as specified with the `rate` option).
- `prg=$p`. Set the progression order to `$p`. The argument `$p` must have one of the following values:
 - `lrpc`. layer-resolution-component-position (LRCP) progressive (i.e., rate scalable)
 - `rlcp`. resolution-layer-component-position (RLCP) progressive (i.e., resolution scalable)
 - `rpcl`. resolution-position-component-layer (RPCL) progressive
 - `pcrl`. position-component-resolution-layer (PCRL) progressive
 - `cpri`. component-position-resolution-layer (CPRL) progressive

By default, LRCP progressive ordering is employed. Note that the RPCL and PCRL progressions are not valid for all possible image geometries. (See [?] for more details.)

- `nomct`. Disallow the use of any multicomponent transform.
- `numrlvls=$n`. Set the number of resolution levels to `$n`. The argument `$n` must be an integer that is greater than or equal to one. The default value is 6.
- `sop`. Generate SOP marker segments.
- `eph`. Generate EPH marker segments.
- `lazy`. Enable lazy coding mode (a.k.a. arithmetic coding bypass).
- `termall`. Terminate all coding passes.
- `segsym`. Use segmentation symbols.
- `vcausal`. Use vertically stripe causal contexts.
- `pterm`. Use predictable termination.
- `resetprob`. Reset the probability models after each coding pass.
- `numgbits=$n`. Set the number of guard bits to `$n`.
- `_jp2overhead=$n`. This is for internal use only. It allows the rate to compensate for the overhead of a container format in which the JPEG-2000 code stream is embedded.

7.5.2 Decoder Options

The JPC decoder supports the following special options:

- `maxlyrs=$n`. Set the maximum number of layers to decode to `$n`.
- `maxpkts=$n`. Set the maximum number of packets to decode to `$n`.

The following generic options are supported:

- `debug`
- `max_samples`

7.5.3 Rate Specification

All rates are specified in terms of compression factors (i.e., as reciprocals of compression ratio) and not as actual bit rates! Although image coding folks frequently use the number of bits per pixel to specify rate, this quantity is often inconvenient to use when dealing with images that have differing sample precisions. Furthermore, the number of bits per pixel is not well defined for multicomponent images with distinct subsampling factors. The compression factor, however, is independent of sample precision and well defined for all types of images. For these reasons, JasPer uses the compression factor and not the number of bits per pixel to specify rates.

7.6 JPG Codec

For lossy coding, one of the most popular image formats is specified in the JPEG standard (i.e., ISO/IEC 10918-1 [?]). In JasPer, the JPG codec implements this format.

The JPEG support in JasPer requires the JPEG library from the Independent JPEG Group (IJG). For legal reasons, the IJG JPEG library source code is not included with JasPer. The source code for this library can be downloaded from the IJG web site (i.e., <http://www.iwg.org>).

7.6.1 Encoder Options

The JPG encoder supports the following special options:

- `quality=$q`. Set the quality factor to `$q`. This is used in order to indirectly control the bit rate for lossy coding.

The JPG encoder does not support any generic options.

7.6.2 Decoder Options

The JPG decoder supports the following special options:

- `version`. Indicate that the decoder should only print the version of the IJG JPEG library in use. (No decoding is performed.)

The JPG decoder supports the following generic options:

- `max_samples`

7.7 PGX Codec

The JPEG-2000 Verification Model software employs a non-standard format called PGX. In JasPer, this format is handled by the PGX codec. The PGX format can only handle single components images, and consequently, is of limited use.

7.7.1 Encoder Options

The PGX encoder does not support any generic or special options.

7.7.2 Decoder Options

The PGX decoder supports the following special options:

- `allow_trunc`. Allow decoding of truncated bitstreams (i.e., reaching EOF during decoding is not deemed to be an error).

The PGX decoder supports the following generic options:

- `max_samples`

7.8 MIF Codec

The MIF format is not a standard format. This format was invented solely for the purpose of testing the JasPer software. The support for the MIF format is experimental. It is intended to be used for advanced testing of the JasPer JPEG-2000 codec implementation. **It is strongly recommended that this codec not be enabled/used in code running in environments where security is a concern.**

7.8.1 Encoder Options

The MIF encoder does not support any generic or special options.

7.8.2 Decoder Options

The MIF decoder does not support any generic or special options.

7.9 PNM Codec

On UNIX platforms, the Portable Pixmap/Graymap/Bitmap (PNM) format is quite popular for coding image data. The PNM codec in JasPer supports this format. In JasPer, the support for the PNM/PGM/PPM format is complete. Therefore, the use of this format is favored over the BMP format. A (nonstandard) extension has also been added to the support for the PNM format so that it can handle images with signed sample values.

7.9.1 Encoder Options

The PNM encoder supports the following special options:

- `text`. Use a non-raw (i.e., non-binary) flavor of PNM format.

7.9.2 Decoder Options

The PNM decoder supports the following special options:

- `allow_trunc`. Allow decoding of truncated bitstreams (i.e., reaching EOF during decoding is not deemed to be an error).

The PNM decoder supports the following generic options:

- `max_samples`

7.10 RAS Codec

One popular image format on Sun workstations is the Sun Rasterfile format. The RAS codec in JasPer implements this format.

7.10.1 Encoder Options

The encoder does not support any special options.

7.10.2 Decoder Options

The RAS decoder supports the following special options:

- `allow_trunc`. Allow decoding of truncated bitstreams (i.e., reaching EOF during decoding is not deemed to be an error).

The RAS decoder supports the following generic options:

- `max_samples`

Chapter 8

Bug Tracking and Reporting

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 page for Jasper on GitHub, which can be found at:

- <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. To submit a bug report, visit the above issue-tracker page, and click on the "New issue" button.

In order to ensure that your bug report can be properly processed, always be sure to include **all** of the following information:

- The version of Jasper in which the problem was found.
- The details of the run-time system (i.e., operating system, version number).
- The compiler that you are using (i.e., vendor, version number).
- The exact command line options used when the problem was observed.
- Indicate whether or not the problem is reproducible. If the problem is reproducible, indicate the exact steps required to reproduce the problem.
- A detailed description of the problem that you are experiencing.

It is essential that you include all of the above information. Failure to do so may result in the bug report not being processed.

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 9

Deprecated List

Member [jas_cleanup](#) (void)

This function is deprecated.

Member [jas_getdbglevel](#) (void)

This function is deprecated.

Member [jas_init](#) (void)

This function is deprecated.

Member [jas_stream_memopen2](#) (char *buffer, size_t buffer_size)

Do not use this function. This function is deprecated. Use [jas_stream_memopen](#) instead.

Chapter 10

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`, `ssize_t count`)

TODO/FIXME: should return type be `ssize_t` and the return value be the count of the characters copied? Perhaps, it might be safer to introduce a new function with differing semantics and deprecate this one?

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_length` (`jas_stream_t *stream`)

Should the return type be long or `ssize_t`? long is consistent with the type used for seek offsets.

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_read` (`jas_stream_t *stream`, `void *buffer`, `size_t count`)

TODO: should `jas_stream_error` be true if `RWLIMIT` exceeded? or maybe introduce a `jas_stream_rwlimit` predicate?

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

TODO/FIXME: Should this macro evaluate to void?

Chapter 11

Topic Index

11.1 Topics

Here is a list of all topics with brief descriptions:

Configuration, Initialization, and Shutdown	??
Memory Allocation	??
I/O Streams	??
Image Representation	??
Color Management	??
One- and Two-Dimensional Sequences	??
Fixed-Point Arithmetic	??
Logging	??
Timers	??
Command-Line Interface (CLI) Option Processing	??
Tag-Value Pair (TVP) Parsing	??
String Processing	??

Chapter 12

Class Index

12.1 Class List

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

jas_allocator_s	A memory allocator	??
jas_cmclrspcconv_t	Color space conversion	??
jas_cmcmpfmt_t	Component format	??
jas_cmpixmap_t	Pixmap (i.e., multicomponent) format	??
jas_cmpprof_t		??
jas_cmpxform_s	Transform class	??
jas_cmpxformops_t	Transform operations	??
jas_cmpxformseq_t	Primitive transform sequence class	??
jas_cmshaplut_t	Shaper look-up table (LUT)	??
jas_cmshapmat_t	Shaper matrix	??
jas_cmshapmatlut_t	Shaper matrix look-up table (LUT)	??
jas_cmxform_t	Primitive transform class	??
jas_image_cmpt_t	Image component class	??
jas_image_cmptparm_t	Component parameters class	??
jas_image_fmt_t	Entry in image format table	??
jas_image_fmtnfo_t	Image format information	??

jas_image_fmtops_t	Image format-dependent operations	??
jas_image_t	Image class	??
jas_logtype_t	Type used for the log type	??
jas_matrix_t	Matrix type	??
jas_opt_t	Command line option type	??
jas_seq2d_t	Two-dimensional sequence type	??
jas_seq_t	One-dimensional sequence type	??
jas_std_allocator_t	The standard library allocator (i.e., a wrapper for malloc and friends)	??
jas_stream_t	I/O stream object	??
jas_taginfo_t	Tag information type	??
jas_tmr_t	Timer type	??
jas_tvparser_t	Tag-value parser type	??

Chapter 13

File Index

13.1 File List

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

bmp_cod.h	??
bmp_enc.h	??
jas_cm.h		
JasPer Color Management	??
jas_compiler.h		
Compiler-related macros	??
jas_debug.h		
JasPer Debugging-Related Functionality	??
jas_dll.h		
Shared Library Macros	??
jas_fix.h		
JasPer Fixed-Point Number Class	??
jas_getopt.h		
Command Line Option Parsing Code	??
jas_icc.h		
ICC Profile	??
jas_image.h		
JasPer Image Class	??
jas_init.h		
JasPer Initialization/Cleanup Code	??
jas_log.h		
JasPer Logging Functionality	??
jas_malloc.h		
JasPer Memory Allocator	??
jas_math.h		
Math-Related Code	??
jas_seq.h		
Sequence/Matrix Library	??
jas_stream.h		
I/O Stream Class	??
jas_string.h		
String Library	??

jas_thread.h	
Threads	??
jas_tmr.h	
Timer Code	??
jas_tvp.h	
Tag/Value Pair Parser	??
jas_types.h	
Primitive Types	??
jas_version.h	
JasPer Version	??
jasper.h	
JasPer Main Header	??
jp2_cod.h	??
jp2_dec.h	??
jpc_bs.h	??
jpc_cod.h	??
jpc_cs.h	??
jpc_dec.h	??
jpc_enc.h	??
jpc_fix.h	??
jpcflt.h	??
jpc_math.h	??
jpc_mct.h	??
jpc_mqcod.h	??
jpc_mqdec.h	??
jpc_mqenc.h	??
jpc_qmfb.h	??
jpc_t1cod.h	??
jpc_t1dec.h	??
jpc_t1enc.h	??
jpc_t2cod.h	??
jpc_t2dec.h	??
jpc_t2enc.h	??
jpc_tagtree.h	??
jpc_tsfb.h	??
jpc_util.h	??
jpg_cod.h	??
jpg_enc.h	??
jpg_jpeglib.h	??
mif_cod.h	??
pgx_cod.h	??
pgx_enc.h	??
pnm_cod.h	??
pnm_enc.h	??
ras_cod.h	??
ras_enc.h	??

Chapter 14

Topic Documentation

14.1 Configuration, Initialization, and Shutdown

Configuration, Initialization, and Shutdown.

Typedefs

- typedef void * [jas_context_t](#)
An opaque handle type used to represent a Jasper library context.

Functions

- JAS_EXPORT void [jas_conf_clear](#) (void)
Configure the Jasper library with the default configuration settings.
- JAS_EXPORT void [jas_conf_set_multithread](#) (int multithread)
Set the multithreading flag for the library.
- JAS_EXPORT void [jas_conf_set_allocator](#) ([jas_allocator_t](#) *allocator)
Set the memory allocator to be used by the library.
- JAS_EXPORT void [jas_conf_set_debug_level](#) (int debug_level)
Set the initial debug level for the library.
- JAS_EXPORT void [jas_conf_set_max_mem_usage](#) (size_t max_mem)
Set the maximum amount of memory that can be used by the library (assuming the allocator wrapper is not disabled).
- JAS_EXPORT void [jas_conf_set_dec_default_max_samples](#) (size_t max_samples)
Set the default value for the maximum number of samples that is allowed in an image to be decoded.
- JAS_EXPORT void [jas_conf_set_vlogmsgf](#) ([jas_vlogmsgf_t](#) *func)
Set the function used by the library to output error, warning, and informational messages.
- JAS_EXPORT int [jas_init_library](#) (void)
Initialize the Jasper library with the current configuration settings.
- JAS_EXPORT int [jas_cleanup_library](#) (void)
Perform clean up for the Jasper library.
- JAS_EXPORT int [jas_init_thread](#) (void)

- Perform per-thread initialization for the JasPer library.*

 - JAS_EXPORT int [jas_cleanup_thread](#) (void)
- Perform per-thread cleanup for the JasPer library.*

 - JAS_EXPORT int [jas_init](#) (void)
- Configure and initialize the JasPer library using the default configuration settings.*

 - JAS_EXPORT void [jas_cleanup](#) (void)
- Perform any clean up for the JasPer library.*

 - JAS_EXPORT [jas_context_t](#) [jas_context_create](#) (void)
- Create a context.*

 - JAS_EXPORT void [jas_context_destroy](#) ([jas_context_t](#) context)
- Destroy a context.*

 - JAS_EXPORT [jas_context_t](#) [jas_get_default_context](#) (void)
- Get the current context for the calling thread.*

 - JAS_EXPORT [jas_context_t](#) [jas_get_context](#) (void)
- Get the current context for the calling thread.*

 - JAS_EXPORT void [jas_set_context](#) ([jas_context_t](#) context)
- Set the current context for the calling thread.*

 - JAS_EXPORT void [jas_set_debug_level](#) (int debug_level)
- Set the debug level for a particular context.*

 - static int [jas_get_debug_level](#) (void)
- Get the debug level for a particular context.*

 - JAS_EXPORT void [jas_set_dec_default_max_samples](#) (size_t max_samples)
- Set the default maximum number of samples that a decoder is permitted to process.*

 - static size_t [jas_get_dec_default_max_samples](#) (void)
- Get the default maximum number of samples that a decoder is permitted to process.*

 - JAS_EXPORT void [jas_set_vlogmsgf](#) ([jas_vlogmsgf_t](#) *func)
- Set the function to be used for log messages.*

 - static [jas_vlogmsgf_t](#) * [jas_get_vlogmsgf](#) (void)
- Get the function to be used for log messages.*

14.1.1 Detailed Description

Configuration, Initialization, and Shutdown.

General information can be found [here](#).

14.1.2 Typedef Documentation

14.1.2.1 [jas_context_t](#)

```
typedef void* jas\_context\_t
```

An opaque handle type used to represent a JasPer library context.

14.1.3 Function Documentation

14.1.3.1 `jas_cleanup()`

```
JAS_EXPORT void jas_cleanup (  
    void )
```

Perform any clean up for the Jasper library.

This function performs any clean up for the Jasper library.

Warning

Configuration, initialization, and cleanup of the library must be performed on the same thread.

Deprecated This function is deprecated.

14.1.3.2 `jas_cleanup_library()`

```
JAS_EXPORT int jas_cleanup_library (  
    void )
```

Perform clean up for the Jasper library.

At the point when this function is called, all threads that have called [jas_init_thread\(\)](#) must have called [jas_cleanup_thread\(\)](#).

Returns

If the operation is successful, zero is returned. Otherwise, a nonzero value is returned.

14.1.3.3 `jas_cleanup_thread()`

```
JAS_EXPORT int jas_cleanup_thread (  
    void )
```

Perform per-thread cleanup for the Jasper library.

14.1.3.4 `jas_conf_clear()`

```
JAS_EXPORT void jas_conf_clear (  
    void )
```

Configure the Jasper library with the default configuration settings.

This function configures the Jasper library with the default configuration settings. These settings may be change via the `jas_conf_*` family of function prior to invoking [jas_init_library\(\)](#).

Warning

Configuration, initialization, and cleanup of the library must be performed on the same thread.

14.1.3.5 `jas_conf_set_allocator()`

```
JAS_EXPORT void jas_conf_set_allocator (  
    jas_allocator_t * allocator)
```

Set the memory allocator to be used by the library.

The object referenced by `allocator` must have a live at least until `jas_cleanup()` is invoked. How the memory in which `*allocator` reside is allocated is the responsibility of the caller.

14.1.3.6 `jas_conf_set_debug_level()`

```
JAS_EXPORT void jas_conf_set_debug_level (  
    int debug_level)
```

Set the initial debug level for the library.

14.1.3.7 `jas_conf_set_dec_default_max_samples()`

```
JAS_EXPORT void jas_conf_set_dec_default_max_samples (  
    size_t max_samples)
```

Set the default value for the maximum number of samples that is allowed in an image to be decoded.

14.1.3.8 `jas_conf_set_max_mem_usage()`

```
JAS_EXPORT void jas_conf_set_max_mem_usage (  
    size_t max_mem)
```

Set the maximum amount of memory that can be used by the library (assuming the allocator wrapper is not disabled).

Warning

It is strongly recommended that the memory usage limit not be set to an excessively large value, as this poses security risks (e.g., decoding a malicious image file could exhaust all virtual memory and effectively crash the system).

14.1.3.9 `jas_conf_set_multithread()`

```
JAS_EXPORT void jas_conf_set_multithread (  
    int multithread)
```

Set the multithreading flag for the library.

14.1.3.10 `jas_conf_set_vlogmsgf()`

```
JAS_EXPORT void jas_conf_set_vlogmsgf (
    jas_vlogmsgf_t * func)
```

Set the function used by the library to output error, warning, and informational messages.

14.1.3.11 `jas_context_create()`

```
JAS_EXPORT jas_context_t jas_context_create (
    void )
```

Create a context.

14.1.3.12 `jas_context_destroy()`

```
JAS_EXPORT void jas_context_destroy (
    jas_context_t context)
```

Destroy a context.

The context being destroyed must not be the current context.

14.1.3.13 `jas_get_context()`

```
JAS_EXPORT jas_context_t jas_get_context (
    void )
```

Get the current context for the calling thread.

14.1.3.14 `jas_get_debug_level()`

```
int jas_get_debug_level (
    void ) [inline], [static]
```

Get the debug level for a particular context.

14.1.3.15 `jas_get_dec_default_max_samples()`

```
size_t jas_get_dec_default_max_samples (
    void ) [inline], [static]
```

Get the default maximum number of samples that a decoder is permitted to process.

14.1.3.16 `jas_get_default_context()`

```
JAS_EXPORT jas\_context\_t jas_get_default_context (
    void )
```

Get the current context for the calling thread.

14.1.3.17 `jas_get_vlogmsgf()`

```
jas\_vlogmsgf\_t * jas_get_vlogmsgf (
    void ) [inline], [static]
```

Get the function to be used for log messages.

14.1.3.18 `jas_init()`

```
JAS_EXPORT int jas_init (
    void )
```

Configure and initialize the Jasper library using the default configuration settings.

The [jas_init\(\)](#) function initializes the Jasper library. The library must be initialized before most code in the library can be used.

The [jas_init\(\)](#) function exists only for reasons of backward compatibility with earlier versions of the library. It is recommended that this function not be used. Instead, the [jas_conf_clear\(\)](#) and [jas_init_library\(\)](#) functions should be used to configure and initialize the library.

Returns

If the library is successfully initialized, zero is returned; otherwise, a nonzero value is returned.

Warning

Configuration, initialization, and cleanup of the library must be performed on the same thread.

Deprecated This function is deprecated.

14.1.3.19 `jas_init_library()`

```
JAS_EXPORT int jas_init_library (
    void )
```

Initialize the Jasper library with the current configuration settings.

The library must be configured by invoking the [jas_conf_clear\(\)](#) function prior to calling [jas_init_library\(\)](#).

Returns

If the initialization of the library is successful, zero is returned; otherwise, a nonzero value is returned.

Warning

The [jas_init_library\(\)](#) function does NOT synchronize with the [jas_conf_clear\(\)](#) function. Configuration, initialization, and cleanup of the library must be performed on the same thread.

14.1.3.20 `jas_init_thread()`

```
JAS_EXPORT int jas_init_thread (
    void )
```

Perform per-thread initialization for the JasPer library.

The library must be initialized by invoking the `jas_init_library()` function prior to calling `jas_init_thread()`.

Warning

The `jas_init_thread()` function can only be invoked in a single thread unless the run-time configuration has enabled multithreading via `jas_set_multithread()`.

14.1.3.21 `jas_set_context()`

```
JAS_EXPORT void jas_set_context (
    jas_context_t context)
```

Set the current context for the calling thread.

14.1.3.22 `jas_set_debug_level()`

```
JAS_EXPORT void jas_set_debug_level (
    int debug_level)
```

Set the debug level for a particular context.

14.1.3.23 `jas_set_dec_default_max_samples()`

```
JAS_EXPORT void jas_set_dec_default_max_samples (
    size_t max_samples)
```

Set the default maximum number of samples that a decoder is permitted to process.

14.1.3.24 `jas_set_vlogmsgf()`

```
JAS_EXPORT void jas_set_vlogmsgf (
    jas_vlogmsgf_t * func)
```

Set the function to be used for log messages.

14.2 Memory Allocation

Memory Allocation.

Classes

- struct [jas_allocator_s](#)
A memory allocator.
- struct [jas_std_allocator_t](#)
The standard library allocator (i.e., a wrapper for malloc and friends).

Typedefs

- typedef struct [jas_allocator_s](#) [jas_allocator_t](#)
A memory allocator.

Functions

- JAS_EXPORT void * [jas_malloc](#) (size_t size)
Allocate memory.
- JAS_EXPORT void [jas_free](#) (void *ptr)
Free memory.
- JAS_EXPORT void * [jas_realloc](#) (void *ptr, size_t size)
Resize a block of allocated memory.
- JAS_EXPORT void * [jas_calloc](#) (size_t num_elements, size_t element_size)
Allocate a block of memory and initialize the contents to zero.
- JAS_EXPORT void * [jas_alloc2](#) (size_t num_elements, size_t element_size)
Allocate array (with overflow checking).
- JAS_EXPORT void * [jas_alloc3](#) (size_t num_arrays, size_t array_size, size_t element_size)
Allocate array of arrays (with overflow checking).
- JAS_EXPORT void * [jas_realloc2](#) (void *ptr, size_t num_elements, size_t element_size)
Resize a block of allocated memory (with overflow checking).
- JAS_EXPORT void [jas_set_max_mem_usage](#) (size_t max_mem)
Set the maximum memory usage allowed by the allocator wrapper.
- JAS_EXPORT size_t [jas_get_mem_usage](#) (void)
Get the current memory usage from the allocator wrapper.
- JAS_EXPORT void [jas_std_allocator_init](#) ([jas_std_allocator_t](#) *allocator)
Initialize a memory allocator that uses malloc and related functions for managing memory.
- JAS_EXPORT void [jas_allocator_cleanup](#) ([jas_allocator_t](#) *allocator)
Clean up an allocator that is no longer needed.
- JAS_EXPORT size_t [jas_get_total_mem_size](#) (void)
Get the total amount of memory available on the system.

14.2.1 Detailed Description

Memory Allocation.

General information can be found [here](#).

14.2.2 Typedef Documentation

14.2.2.1 `jas_allocator_t`

```
typedef struct jas_allocator_s jas_allocator_t
```

A memory allocator.

14.2.3 Function Documentation

14.2.3.1 `jas_alloc2()`

```
JAS_EXPORT void * jas_alloc2 (  
    size_t num_elements,  
    size_t element_size)
```

Allocate array (with overflow checking).

14.2.3.2 `jas_alloc3()`

```
JAS_EXPORT void * jas_alloc3 (  
    size_t num_arrays,  
    size_t array_size,  
    size_t element_size)
```

Allocate array of arrays (with overflow checking).

14.2.3.3 `jas_allocator_cleanup()`

```
JAS_EXPORT void jas_allocator_cleanup (  
    jas_allocator_t * allocator)
```

Clean up an allocator that is no longer needed.

This function cleans up an allocator, releasing any resources associated with the allocator. After clean up is performed, the allocator can no longer be used.

14.2.3.4 `jas_calloc()`

```
JAS_EXPORT void * jas_calloc (
    size_t num_elements,
    size_t element_size)
```

Allocate a block of memory and initialize the contents to zero.

This function has an identical behavior as `calloc` (from the C standard library).

14.2.3.5 `jas_free()`

```
JAS_EXPORT void jas_free (
    void * ptr)
```

Free memory.

This function has an identical behavior as `free` (from the C standard library).

14.2.3.6 `jas_get_mem_usage()`

```
JAS_EXPORT size_t jas_get_mem_usage (
    void )
```

Get the current memory usage from the allocator wrapper.

This function queries the amount of memory currently in use by the allocator wrapper. This function can only be called if the use of the allocator wrapper is enabled. Calling this function if the allocator wrapper is not enabled results in undefined behavior.

14.2.3.7 `jas_get_total_mem_size()`

```
JAS_EXPORT size_t jas_get_total_mem_size (
    void )
```

Get the total amount of memory available on the system.

This function may be called prior to the library being initialized. In fact, this function may be useful for determining a reasonable value for the memory limit setting to be used during (run-time) library configuration.

Returns

The total amount of memory available (in bytes) is returned, if this can be determined. Otherwise, zero is returned.

14.2.3.8 jas_malloc()

```
JAS_EXPORT void * jas_malloc (
    size_t size)
```

Allocate memory.

This function has an identical behavior as malloc (from the C standard library), except that a zero-sized allocation returns a non-null pointer (assuming no out-of-memory error occurs).

14.2.3.9 jas_realloc()

```
JAS_EXPORT void * jas_realloc (
    void * ptr,
    size_t size)
```

Resize a block of allocated memory.

This function has an identical behavior as realloc (from the C standard library).

14.2.3.10 jas_realloc2()

```
JAS_EXPORT void * jas_realloc2 (
    void * ptr,
    size_t num_elements,
    size_t element_size)
```

Resize a block of allocated memory (with overflow checking).

14.2.3.11 jas_set_max_mem_usage()

```
JAS_EXPORT void jas_set_max_mem_usage (
    size_t max_mem)
```

Set the maximum memory usage allowed by the allocator wrapper.

Parameters

<i>max_mem</i>	The maximum amount of memory (in bytes) that the allocator can use.
----------------	---

This function sets the maximum amount of memory (in bytes) that the allocator wrapper is permitted to use to `max_mem`. If `max_mem` is zero, no limit is imposed on the amount of memory used by allocator. This function can only be called if the use of the allocator wrapper is enabled. Calling this function if the allocator wrapper is not enabled results in undefined behavior. The limit on the amount of memory that the allocator can use should never be set to a value less than the amount of memory currently being used by the allocator (as doing so results in undefined behavior).

14.2.3.12 jas_std_allocator_init()

```
JAS_EXPORT void jas_std_allocator_init (
    jas_std_allocator_t * allocator)
```

Parameters

<i>allocator</i>	A pointer to the storage in memory that will hold the state associated with the allocator.
------------------	--

The object referenced by `allocator` must have a lifetime that extends until `jas_allocator_cleanup` is called for the allocator.

14.3 I/O Streams

I/O streams.

Classes

- struct `jas_stream_t`
I/O stream object.

Macros

- #define `jas_stream_eof(stream)`
Get the EOF indicator for a stream.
- #define `jas_stream_error(stream)`
Get the error indicator for a stream.
- #define `jas_stream_clearerr(stream)`
Clear the error indicator for a stream.
- #define `jas_stream_getrwlmit(stream)`
Get the read/write limit for a stream.
- #define `jas_stream_getrwcoun(stream)`
Get the read/write count for a stream.
- #define `jas_stream_getc(stream)`
jas_stream_getc Read a character from a stream.
- #define `jas_stream_putc(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_EXPORT [jas_stream_t](#) * [jas_stream_fopen](#) (const char *filename, const char *mode)
Open a file as a stream.
- JAS_EXPORT [jas_stream_t](#) * [jas_stream_memopen](#) (char *buffer, size_t buffer_size)
Open a memory buffer as a stream.
- JAS_DEPRECATED JAS_EXPORT [jas_stream_t](#) * [jas_stream_memopen2](#) (char *buffer, size_t buffer_size)
Do not use.
- JAS_EXPORT [jas_stream_t](#) * [jas_stream_fdopen](#) (int fd, const char *mode)
Open a file descriptor as a stream.
- JAS_EXPORT [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_EXPORT [jas_stream_t](#) * [jas_stream_tmpfile](#) (void)
Open a temporary file as a stream.
- JAS_EXPORT int [jas_stream_close](#) ([jas_stream_t](#) *stream)
Close a stream.
- JAS_EXPORT long [jas_stream_setrwlimit](#) ([jas_stream_t](#) *stream, long rwlimit)
Set the read/write limit for a stream.
- JAS_EXPORT long [jas_stream_setrwcoun](#) ([jas_stream_t](#) *stream, long rw_count)
Set the read/write count for a stream.
- JAS_EXPORT size_t [jas_stream_read](#) ([jas_stream_t](#) *stream, void *buffer, size_t count)
Read characters from a stream into a buffer.
- JAS_EXPORT 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_EXPORT size_t [jas_stream_write](#) ([jas_stream_t](#) *stream, const void *buffer, size_t count)
Write characters from a buffer to a stream.
- JAS_EXPORT int [jas_stream_printf](#) ([jas_stream_t](#) *stream, const char *format,...)
Write formatted output to a stream.
- JAS_EXPORT int [jas_stream_puts](#) ([jas_stream_t](#) *stream, const char *s)
Write a string to a stream.
- JAS_EXPORT char * [jas_stream_gets](#) ([jas_stream_t](#) *stream, char *buffer, int buffer_size)
Read a line of input from a stream.
- JAS_EXPORT int [jas_stream_ungetc](#) ([jas_stream_t](#) *stream, int c)
Put a character back on a stream.
- JAS_EXPORT JAS_ATTRIBUTE_PURE int [jas_stream_isseekable](#) ([jas_stream_t](#) *stream)
Determine if stream supports seeking.
- JAS_EXPORT long [jas_stream_seek](#) ([jas_stream_t](#) *stream, long offset, int origin)
Set the current position within the stream.
- JAS_EXPORT long [jas_stream_tell](#) ([jas_stream_t](#) *stream)
Get the current position within the stream.
- JAS_EXPORT int [jas_stream_rewind](#) ([jas_stream_t](#) *stream)
Seek to the beginning of a stream.
- JAS_EXPORT int [jas_stream_flush](#) ([jas_stream_t](#) *stream)
Flush any pending output to a stream.
- JAS_EXPORT int [jas_stream_copy](#) ([jas_stream_t](#) *destination, [jas_stream_t](#) *source, ssize_t count)
Copy data from one stream to another.

- JAS_EXPORT int [jas_stream_display](#) ([jas_stream_t](#) *stream, FILE *fp, int count)
Print a hex dump of data read from a stream.
- JAS_EXPORT ssize_t [jas_stream_gobble](#) ([jas_stream_t](#) *stream, size_t count)
Consume (i.e., discard) characters from stream.
- JAS_EXPORT ssize_t [jas_stream_pad](#) ([jas_stream_t](#) *stream, size_t count, int value)
Write a fill character multiple times to a stream.
- JAS_EXPORT long [jas_stream_length](#) ([jas_stream_t](#) *stream)
Get the size of the file associated with the specified stream.

14.3.1 Detailed Description

I/O streams.

General information can be found [here](#).

14.3.2 Macro Definition Documentation

14.3.2.1 [jas_stream_clearerr](#)

```
#define jas_stream_clearerr(  
    stream)
```

Value:

```
((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?

14.3.2.2 [jas_stream_eof](#)

```
#define jas_stream_eof(  
    stream)
```

Value:

```
((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.

14.3.2.3 jas_stream_error

```
#define jas_stream_error(  
    stream)
```

Value:

```
((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.

14.3.2.4 jas_stream_getc

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

Value:

```
jas_stream_getc_func(stream)
```

`jas_stream_getc` Read a character from a stream.

Parameters

<i>stream</i>	A pointer to the stream from which to read a character.
---------------	---

Returns

If a character is successfully read, the character is returned. Otherwise, EOF is returned.

14.3.2.5 jas_stream_getwcount

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.

14.3.2.6 jas_stream_getrwlimit

```
#define jas_stream_getrwlimit(  
    stream)
```

Value:

```
((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).

14.3.2.7 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.

14.3.2.8 jas_stream_putc

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

Value:

```
jas_stream_putc_func(stream, c)
```

jas_stream_putc Write a character to a stream.

Parameters

<i>stream</i>	A pointer to the stream to which to write the character.
<i>c</i>	The character to be written.

Returns

If the character is successfully output, the value of the character is returned. Otherwise, EOF is returned.

14.3.3 Function Documentation

14.3.3.1 jas_stream_close()

```
JAS_EXPORT 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.

14.3.3.2 jas_stream_copy()

```
JAS_EXPORT int jas_stream_copy (
    jas_stream_t * destination,
    jas_stream_t * source,
    ssize_t 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. In particular, if *count* is nonnegative, *count* characters are copied from the source stream *source* to the destination stream *destination*. Otherwise (i.e., if *count* is negative), the entire source stream *source* (i.e., until EOF is reached) is copied to the destination stream *destination*.

Returns

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

Todo TODO/FIXME: should return type be `ssize_t` and the return value be the count of the characters copied? Perhaps, it might be safer to introduce a new function with differing semantics and deprecate this one?

14.3.3.3 jas_stream_display()

```
JAS_EXPORT 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?

14.3.3.4 jas_stream_fdopen()

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

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 <code>fdopen</code> function in the C standard library.

Returns

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

14.3.3.5 `jas_stream_flush()`

```
JAS_EXPORT 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.

(This function is analogous to `fflush` for C standard library streams.)

Returns

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

14.3.3.6 `jas_stream_fopen()`

```
JAS_EXPORT 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 <code>fopen</code> function in the C standard library.

Returns

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

14.3.3.7 `jas_stream_freopen()`

```
JAS_EXPORT 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 null-terminated string containing the pathname of the file to be reopened.
<i>mode</i>	A pointer to a null-terminated string containing the mode to be used for reopening the file. This string is similar to that used by the <code>fopen</code> function in the C standard library.
<i>fp</i>	A pointer to the <code>FILE</code> (i.e., stdio stream) to be reopened.

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.

14.3.3.8 `jas_stream_gets()`

```
JAS_EXPORT 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.

(This function is analogous to `fgets` for C standard library streams.)

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.

14.3.3.9 `jas_stream_gobble()`

```
JAS_EXPORT ssize_t jas_stream_gobble (  
    jas\_stream\_t * stream,  
    size_t 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.

14.3.3.10 `jas_stream_isseekable()`

```
JAS_EXPORT JAS_ATTRIBUTE_PURE 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.

14.3.3.11 `jas_stream_length()`

```
JAS_EXPORT long jas_stream_length (  
    jas\_stream\_t * stream)
```

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

Parameters

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

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.

Todo Should the return type be long or ssize_t? long is consistent with the type used for seek offsets.

14.3.3.12 jas_stream_memopen()

```
JAS_EXPORT jas_stream_t * jas_stream_memopen (
    char * buffer,
    size_t 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*.

14.3.3.13 jas_stream_memopen2()

```
JAS_DEPRECATED JAS_EXPORT jas_stream_t * jas_stream_memopen2 (
    char * buffer,
    size_t buffer_size)
```

Do not use.

Deprecated Do not use this function. This function is deprecated. Use *jas_stream_memopen* instead.

14.3.3.14 jas_stream_pad()

```
JAS_EXPORT ssize_t jas_stream_pad (
```

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.

14.3.3.15 jas_stream_peek()

```
JAS_EXPORT 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; also should return type be changed to `size_t`?

14.3.3.16 jas_stream_printf()

```
JAS_EXPORT 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.

14.3.3.17 jas_stream_puts()

```
JAS_EXPORT 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.

(This function is analogous to fputs for C standard library streams.)

Returns

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

14.3.3.18 jas_stream_read()

```
JAS_EXPORT size_t jas_stream_read (  
    jas_stream_t * stream,  
    void * buffer,  
    size_t 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.

(This function is analogous to `fread` with the two read-count parameters combined into a single size.)

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.

(The functions `jas_stream_getrwcoun`[t\(\)](#) and `jas_stream_getrwlimit()` can be used to distinguish between failure due to an I/O error and failure due to the read/write limit being exceeded.)

Todo TODO: should `jas_stream_error` be true if `RWLIMIT` exceeded? or maybe introduce a `jas_stream_rwlimit` predicate?

14.3.3.19 `jas_stream_rewind()`

```
JAS_EXPORT 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)`.

(This function is analogous to `frewind` for C standard library streams.)

Returns

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

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.

14.3.3.21 jas_stream_setrwcoun()

```
JAS_EXPORT 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?

14.3.3.22 jas_stream_setrwlmit()

```
JAS_EXPORT long jas_stream_setrwlmit (  
    jas_stream_t * stream,  
    long rwlmit)
```

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.

14.3.3.23 `jas_stream_tell()`

```
JAS_EXPORT 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.

(This function is analogous to `ftell` for C standard library streams.)

Returns

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

14.3.3.24 `jas_stream_tmpfile()`

```
JAS_EXPORT 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.

14.3.3.25 `jas_stream_ungetc()`

Generated by Doxygen

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

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.

14.3.3.26 jas_stream_write()

```
JAS_EXPORT size_t jas_stream_write (
    jas_stream_t * stream,
    const void * buffer,
    size_t 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.

(This function is analogous to `fwrite` with the two write-count parameters combined into a single size.)

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

14.4 Image Representation

Image Representation.

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_fmtdinfo_t](#)
Image format information.

Macros

- #define [JAS_IMAGE_MAXFMTS](#) 32
The maximum number of image data formats supported.
- #define [jas_image_width](#)(image)
Get the width of the image in units of the image reference grid.
- #define [jas_image_height](#)(image)
Get the height of the image in units of the image reference grid.
- #define [jas_image_tlx](#)(image)
Get the x-coordinate of the top-left corner of the image bounding box on the reference grid.
- #define [jas_image_tly](#)(image)
Get the y-coordinate of the top-left corner of the image bounding box on the reference grid.
- #define [jas_image_brx](#)(image)
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)
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)
Get the number of image components.
- #define [jas_image_clrspc](#)(image)
Get the color model used by the image.
- #define [jas_image_setclrspc](#)(image, clrspc)
Set the color model for an image.
- #define [jas_image_cmptwidth](#)(image, cmptno)
Get the width of a component.
- #define [jas_image_cmptheight](#)(image, cmptno)
Get the height of a component.
- #define [jas_image_cmptsgnd](#)(image, cmptno)
Get the signedness of the sample data for a component.
- #define [jas_image_cmptprec](#)(image, cmptno)
Get the precision of the sample data for a component.
- #define [jas_image_cmptstep](#)(image, cmptno)
Get the horizontal subsampling factor for a component.
- #define [jas_image_cmptvstep](#)(image, cmptno)
Get the vertical subsampling factor for a component.

- `#define jas_image_cmpttlx(image, cmptno)`
Get the x-coordinate of the top-left corner of a component.
- `#define jas_image_cmpttly(image, cmptno)`
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)`
Get the color management profile of an image.
- `#define jas_image_setcmprofn(image, 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_EXPORT jas_image_t * jas_image_create (unsigned numcmpts, const jas_image_cmptparm_t *cmptparms, jas_colorspc_t clrspc)`
Create an image.
- `JAS_EXPORT jas_image_t * jas_image_create0 (void)`
Create an "empty" image.
- `JAS_EXPORT jas_image_t * jas_image_copy (jas_image_t *image)`
Clone an image.
- `JAS_EXPORT void jas_image_destroy (jas_image_t *image)`
Deallocate any resources associated with an image.
- `JAS_ATTRIBUTE_PURE JAS_EXPORT 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_EXPORT 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_EXPORT 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_EXPORT 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_EXPORT 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_EXPORT 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_EXPORT void [jas_image_delcmpt](#) ([jas_image_t](#) *image, unsigned cmptno)
Delete a component from an image.
- JAS_EXPORT int [jas_image_addcmpt](#) ([jas_image_t](#) *image, int cmptno, const [jas_image_cmptparm_t](#) *cmptparm)
Add a component to an image.
- JAS_EXPORT 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_EXPORT 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_EXPORT int [jas_image_readcmptsample](#) ([jas_image_t](#) *image, unsigned cmptno, unsigned x, unsigned y)
Read a component sample for an image.
- JAS_EXPORT 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_EXPORT int [jas_image_getcmptbytype](#) (const [jas_image_t](#) *image, [jas_image_cmpttype_t](#) ctype)
Get an image component by its type.
- JAS_EXPORT void [jas_image_clearfmts](#) (void)
Clear the table of image formats.
- JAS_EXPORT const [jas_image_fmtinfo_t](#) * [jas_image_getfmtbyind](#) (int index)
Get a image format entry by its table index.
- JAS_EXPORT int [jas_image_getnumfmts](#) (void)
Get the number of image format table entries.
- JAS_EXPORT int [jas_image_setfmtenable](#) (int index, int enabled)
Get the number of image format table entries.
- JAS_EXPORT 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_EXPORT int [jas_image_strtofmt](#) (const char *s)
Get the ID for the image format with the specified name.
- JAS_ATTRIBUTE_CONST JAS_EXPORT const char * [jas_image_fmtostr](#) (int fmt)
Get the name of the image format with the specified ID.
- JAS_ATTRIBUTE_CONST JAS_EXPORT const [jas_image_fmtinfo_t](#) * [jas_image_lookupfmtbyid](#) (int id)
Lookup image format information by the format ID.
- JAS_ATTRIBUTE_PURE JAS_EXPORT const [jas_image_fmtinfo_t](#) * [jas_image_lookupfmtbyname](#) (const char *name)
Lookup image format information by the format name.
- JAS_ATTRIBUTE_PURE JAS_EXPORT int [jas_image_fmtofilename](#) (const char *filename)
Guess the format of an image file based on its name.
- JAS_ATTRIBUTE_PURE JAS_EXPORT int [jas_image_getfmt](#) ([jas_stream_t](#) *in)
Get the format of image data in a stream.
- JAS_ATTRIBUTE_PURE JAS_EXPORT int [jas_image_ishomosamp](#) (const [jas_image_t](#) *image)
Test if the sampling of the image is homogeneous.

- JAS_EXPORT 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_EXPORT 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_EXPORT 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_EXPORT `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.
- JAS_EXPORT int `jas_image_dump` (`jas_image_t` *image, FILE *out)
Dump the information for an image (for debugging).

14.4.1 Detailed Description

Image Representation.

General information can be found [here](#).

14.4.2 Macro Definition Documentation

14.4.2.1 `jas_image_brx`

```
#define jas_image_brx(  
    image)
```

Value:

```
((image)->brx_)
```

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

14.4.2.2 `jas_image_bry`

```
#define jas_image_bry(  
    image)
```

Value:

```
((image)->bry_)
```

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

14.4.2.3 jas_image_clrspc

```
#define jas_image_clrspc(  
    image)
```

Value:

```
((image)->clrspc_)
```

Get the color model used by the image.

14.4.2.4 jas_image_cmprof

```
#define jas_image_cmprof(  
    image)
```

Value:

```
((image)->cmprof_)
```

Get the color management profile of an image.

14.4.2.5 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").

14.4.2.6 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").

14.4.2.7 jas_image_cmptheight

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

Value:

```
((image)->cmpts_[cmptno]->height_)
```

Get the height of a component.

14.4.2.8 jas_image_cmptstep

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

Value:

```
((image)->cmpts_[cmptno]->hstep_)
```

Get the horizontal subsampling factor for a component.

14.4.2.9 jas_image_cmptprec

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

Value:

```
((image)->cmpts_[cmptno]->prec_)
```

Get the precision of the sample data for a component.

14.4.2.10 jas_image_cmptsgnd

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

Value:

```
((image)->cmpts_[cmptno]->sgnd_)
```

Get the signedness of the sample data for a component.

14.4.2.11 jas_image_cmpttlx

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

Value:

```
((image)->cmpts_[cmptno]->tlx_)
```

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

14.4.2.12 jas_image_cmpttly

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

Value:

```
((image)->cmpts_[cmptno]->tly_)
```

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

14.4.2.13 jas_image_cmptvstep

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

Value:

```
((image)->cmpts_[cmptno]->vstep_)
```

Get the vertical subsampling factor for a component.

14.4.2.14 jas_image_cmptwidth

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

Value:

```
((image)->cmpts_[cmptno]->width_)
```

Get the width of a component.

14.4.2.15 jas_image_height

```
#define jas_image_height(  
    image)
```

Value:

```
((image)->bry_ - (image)->tly_)
```

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

14.4.2.16 JAS_IMAGE_MAXFMTS

```
#define JAS_IMAGE_MAXFMTS 32
```

The maximum number of image data formats supported.

14.4.2.17 jas_image_numcmpts

```
#define jas_image_numcmpts(  
    image)
```

Value:

```
((image)->numcmpts_)
```

Get the number of image components.

14.4.2.18 jas_image_setclrspc

```
#define jas_image_setclrspc(  
    image,  
    clrspc)
```

Value:

```
((image)->clrspc_ = (clrspc))
```

Set the color model for an image.

14.4.2.19 jas_image_setcmprof

```
#define jas_image_setcmprof(  
    image,  
    cmprof)
```

Value:

```
((image)->cmprof_ = cmprof)
```

Set the color management profile for an image.

14.4.2.20 jas_image_tlx

```
#define jas_image_tlx(  
    image)
```

Value:

```
((image)->tlx_)
```

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

14.4.2.21 jas_image_tly

```
#define jas_image_tly(  
    image)
```

Value:

```
((image)->tly_)
```

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

14.4.2.22 jas_image_width

```
#define jas_image_width(  
    image)
```

Value:

```
((image)->brx_ - (image)->tlx_)
```

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

14.4.3 Typedef Documentation

14.4.3.1 jas_image_cmpttype_t

```
typedef int_fast32_t jas_image_cmpttype_t
```

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

14.4.3.2 jas_image_colorspc_t

```
typedef int_fast16_t jas_image_colorspc_t
```

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

14.4.3.3 `jas_image_coord_t`

```
typedef int_fast32_t jas_image_coord_t
```

Image coordinate.

14.4.3.4 `jas_image_smplytype_t`

```
typedef int_fast16_t jas_image_smplytype_t
```

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

14.4.4 Function Documentation

14.4.4.1 `jas_image_addcmpt()`

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

Add a component to an image.

14.4.4.2 `jas_image_addfmt()`

```
JAS_EXPORT 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.

14.4.4.3 `jas_image_chclrspc()`

```
JAS_EXPORT 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.

14.4.4.4 `jas_image_clearfmts()`

```
JAS_EXPORT void jas_image_clearfmts (  
    void )
```

Clear the table of image formats.

14.4.4.5 `jas_image_cmpt_domains_same()`

```
JAS_ATTRIBUTE_PURE JAS_EXPORT bool jas_image_cmpt_domains_same (  
    const jas\_image\_t * image)
```

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

14.4.4.6 `jas_image_copy()`

```
JAS_EXPORT jas\_image\_t * jas_image_copy (  
    jas\_image\_t * image)
```

Clone an image.

14.4.4.7 `jas_image_copycmpt()`

```
JAS_EXPORT int jas_image_copycmpt (  
    jas\_image\_t * dstimage,  
    unsigned dstcmptno,  
    jas\_image\_t * srcimage,  
    unsigned srccmptno)
```

Copy a component from one image to another.

14.4.4.8 `jas_image_create()`

```
JAS_EXPORT jas\_image\_t * jas_image_create (  
    unsigned numcmpts,  
    const jas\_image\_cmptparm\_t * cmptparms,  
    jas\_clrspc\_t clrspc)
```

Create an image.

14.4.4.9 `jas_image_create0()`

```
JAS_EXPORT jas\_image\_t * jas_image_create0 (  
    void )
```

Create an "empty" image.

14.4.4.10 `jas_image_decode()`

```
JAS_EXPORT 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.

14.4.4.11 `jas_image_delcmt()`

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

Delete a component from an image.

14.4.4.12 `jas_image_depalettize()`

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

Depalettize an image.

14.4.4.13 `jas_image_destroy()`

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

Deallocate any resources associated with an image.

14.4.4.14 `jas_image_dump()`

```
JAS_EXPORT int jas_image_dump (  
    jas_image_t * image,  
    FILE * out)
```

Dump the information for an image (for debugging).

14.4.4.15 jas_image_encode()

```
JAS_EXPORT 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.

14.4.4.16 jas_image_fmtfromname()

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

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

14.4.4.17 jas_image_fmtostr()

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

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

14.4.4.18 jas_image_getcmptbytype()

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

Get an image component by its type.

14.4.4.19 jas_image_getfmt()

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

Get the format of image data in a stream.

Note that only enabled codecs are used in determining the image format.

14.4.4.20 jas_image_getfmtbyind()

```
JAS_EXPORT const jas_image_fmtinfo_t * jas_image_getfmtbyind (  
    int index)
```

Get a image format entry by its table index.

14.4.4.21 `jas_image_getnumfmts()`

```
JAS_EXPORT int jas_image_getnumfmts (  
    void )
```

Get the number of image format table entries.

14.4.4.22 `jas_image_ishomosamp()`

```
JAS_ATTRIBUTE_PURE JAS_EXPORT int jas_image_ishomosamp (  
    const jas\_image\_t * image)
```

Test if the sampling of the image is homogeneous.

14.4.4.23 `jas_image_lookupfmtbyid()`

```
JAS_ATTRIBUTE_CONST JAS_EXPORT const jas\_image\_fmtinfo\_t * jas_image_lookupfmtbyid (  
    int id)
```

Lookup image format information by the format ID.

14.4.4.24 `jas_image_lookupfmtbyname()`

```
JAS_ATTRIBUTE_PURE JAS_EXPORT const jas\_image\_fmtinfo\_t * jas_image_lookupfmtbyname (  
    const char * name)
```

Lookup image format information by the format name.

14.4.4.25 `jas_image_rawsize()`

```
JAS_ATTRIBUTE_PURE JAS_EXPORT 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).

14.4.4.26 `jas_image_readcmpt()`

```
JAS_EXPORT 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.

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

14.4.4.27 `jas_image_readcmpt2()`

```
JAS_EXPORT 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.

14.4.4.28 `jas_image_readcmptsample()`

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

Read a component sample for an image.

14.4.4.29 `jas_image_sampcmpt()`

```
JAS_EXPORT 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)
```

???

14.4.4.30 `jas_image_setfmttenable()`

```
JAS_EXPORT int jas_image_setfmttenable (  
    int index,  
    int enabled)
```

Get the number of image format table entries.

Warning

This function may be removed in future versions of the library. Do not rely on it.

14.4.4.31 `jas_image_strtofmt()`

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

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

14.4.4.32 `jas_image_writecmt()`

```
JAS_EXPORT int jas_image_writecmt (  
    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.

14.4.4.33 `jas_image_writecmt2()`

```
JAS_EXPORT 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.

14.4.4.34 `jas_image_writecmptsample()`

```
JAS_EXPORT 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.

14.5 Color Management

Color Management.

Classes

- struct [jas_cmcmptfmt_t](#)
Component format.
- struct [jas_cmpixmap_t](#)
Pixmap (i.e., multicomponent) format.
- struct [jas_cmpxformops_t](#)
Transform operations.
- struct [jas_cmshapmatlut_t](#)
Shaper matrix look-up table (LUT).
- struct [jas_cmshapmat_t](#)
Shaper matrix.
- struct [jas_cmshaplut_t](#)
Shaper look-up table (LUT).
- struct [jas_cmclrspcconv_t](#)
Color space conversion.
- struct [jas_cmpxform_s](#)
Transform class.
- struct [jas_cmpxformseq_t](#)
Primitive transform sequence class.
- struct [jas_cmxfom_t](#)
Primitive transform class.
- struct [jas_cmprof_t](#)

Macros

- #define [JAS_CMIFORM_NUMINTENTS](#) 4
Number of rendering intents.
- #define [jas_clrspc_create](#)(fam, mbr)
Create a color space.
- #define [jas_clrspc_fam](#)(clrspc)
Get the family of a color space.
- #define [jas_clrspc_mbr](#)(clrspc)
Get the (family) member of a color space.
- #define [jas_clrspc_isgeneric](#)(clrspc)
Test if a color space is generic.
- #define [jas_clrspc_isunknown](#)(clrspc)
Test if a color space is unknown.
- #define [JAS_CLRSPC_FAM_UNKNOWN](#) 0
Color space families.
- #define [JAS_CLRSPC_UNKNOWN](#) JAS_CLRSPC_UNKNOWNMASK
Specific color spaces.
- #define [JAS_CLRSPC_GENRGB](#) [jas_clrspc_create](#)(JAS_CLRSPC_FAM_RGB, 0)
Generic color spaces.
- #define [jas_cmprof_clrspc](#)(prof)
Get the color space associated with a color-management profile.

Typedefs

- typedef unsigned [jas_clrspc_t](#)
Color space.
- typedef double [jas_cmreal_t](#)
- typedef struct [jas_cmpxform_s](#) [jas_cmpxform_t](#)
Transform class.

Enumerations

- enum [jas_cmxform_op_t](#)
Transform operations.
- enum [jas_cmxform_intent_t](#)
Rendering intents.
- enum [jas_cmxform_optm_t](#)
Transform optimization.

Functions

- JAS_EXPORT [jas_cmprof_t](#) * [jas_cmprof_createfromiccprof](#) (const [jas_iccprof_t](#) *iccprof)
Create a color-management profile from an ICC profile.
- JAS_EXPORT [jas_cmprof_t](#) * [jas_cmprof_createfromclrspc](#) ([jas_clrspc_t](#) clrspc)
Create a color-management profile from a color space.
- JAS_EXPORT void [jas_cmprof_destroy](#) ([jas_cmprof_t](#) *prof)
Destroy a color-management profile.
- JAS_EXPORT [jas_cmprof_t](#) * [jas_cmprof_copy](#) (const [jas_cmprof_t](#) *prof)
Copy a color-management profile.
- JAS_EXPORT [jas_iccprof_t](#) * [jas_iccprof_createfromcmprof](#) (const [jas_cmprof_t](#) *prof)
Create a ICC profile from a CM profile.
- JAS_EXPORT [jas_cmxform_t](#) * [jas_cmxform_create](#) (const [jas_cmprof_t](#) *inprof, const [jas_cmprof_t](#) *outprof, const [jas_cmprof_t](#) *proofprof, [jas_cmxform_op_t](#) op, [jas_cmxform_intent_t](#) intent, [jas_cmxform_optm_t](#) optimize)
Create a transform from a CM profile.
- JAS_EXPORT void [jas_cmxform_destroy](#) ([jas_cmxform_t](#) *xform)
Destroy a transform.
- JAS_EXPORT int [jas_cmxform_apply](#) (const [jas_cmxform_t](#) *xform, const [jas_cmpixmap_t](#) *in, [jas_cmpixmap_t](#) *out)
Apply a transform to data.
- unsigned [jas_clrspc_numchans](#) ([jas_clrspc_t](#) clrspc)
Get the number of channels associated with a particular color space.
- JAS_EXPORT [jas_iccprof_t](#) * [jas_iccprof_load](#) ([jas_stream_t](#) *in)
Read an ICC profile from a stream.
- JAS_EXPORT int [jas_iccprof_save](#) ([jas_iccprof_t](#) *prof, [jas_stream_t](#) *out)
Write an ICC profile to a stream.
- JAS_EXPORT void [jas_iccprof_destroy](#) ([jas_iccprof_t](#) *prof)
Destroy an ICC profile.

- JAS_ATTRIBUTE_PURE JAS_EXPORT jas_iccattrval_t * [jas_iccprof_getattr](#) (const jas_iccprof_t *prof, jas_iccattrname_t name)
Get an attribute of an ICC profile.
- JAS_EXPORT int [jas_iccprof_setattr](#) (jas_iccprof_t *prof, jas_iccattrname_t name, jas_iccattrval_t *val)
Set an attribute of an ICC profile.
- JAS_EXPORT void [jas_iccprof_dump](#) (const jas_iccprof_t *prof, FILE *out)
Dump an ICC profile to a stream in human-readable format for debugging purposes.
- JAS_EXPORT jas_iccprof_t * [jas_iccprof_copy](#) (const jas_iccprof_t *prof)
Create a copy of an ICC profile.
- JAS_EXPORT int [jas_iccprof_gethdr](#) (const jas_iccprof_t *prof, jas_icchdr_t *hdr)
Get the header for an ICC profile.
- JAS_EXPORT int [jas_iccprof_sethdr](#) (jas_iccprof_t *prof, const jas_icchdr_t *hdr)
Set the header for an ICC profile.
- JAS_EXPORT void [jas_iccattrval_destroy](#) (jas_iccattrval_t *attrval)
Destroy an ICC profile attribute.
- JAS_EXPORT int [jas_iccattrval_allowmodify](#) (jas_iccattrval_t **attrval)
TODO/FIXME.
- JAS_EXPORT jas_iccattrval_t * [jas_iccattrval_clone](#) (jas_iccattrval_t *attrval)
Create a copy of an ICC profile attribute.
- JAS_EXPORT jas_iccattrval_t * [jas_iccattrval_create](#) (jas_iccuint32_t type)
Create an ICC profile attribute.
- JAS_EXPORT void [jas_iccattrtab_dump](#) (const jas_iccattrtab_t *attrtab, FILE *out)
Dump an ICC profile attribute to a stream in human-readable format for debugging purposes.
- JAS_EXPORT jas_iccprof_t * [jas_iccprof_createfrombuf](#) (const jas_uchar *buf, unsigned len)
Create an ICC profile from a buffer in memory.
- JAS_EXPORT jas_iccprof_t * [jas_iccprof_createfromclrspc](#) (unsigned clrspc)
Create an ICC profile from a color space.

14.5.1 Detailed Description

Color Management.

General information can be found [here](#).

14.5.2 Macro Definition Documentation

14.5.2.1 `jas_clrspc_create`

```
#define jas_clrspc_create(  
    fam,  
    mbr)
```

Value:

```
(( (fam) << 8) | (mbr))
```

Create a color space.

14.5.2.2 `jas_clrspc_fam`

```
#define jas_clrspc_fam(  
    clrspc)
```

Value:

```
((clrspc) >> 8)
```

Get the family of a color space.

14.5.2.3 `JAS_CLRSPC_FAM_UNKNOWN`

```
#define JAS_CLRSPC_FAM_UNKNOWN 0
```

Color space families.

14.5.2.4 `JAS_CLRSPC_GENRGB`

```
#define JAS_CLRSPC_GENRGB jas\_clrspc\_create(JAS_CLRSPC_FAM_RGB, 0)
```

Generic color spaces.

14.5.2.5 `jas_clrspc_isgeneric`

```
#define jas_clrspc_isgeneric(  
    clrspc)
```

Value:

```
(!jas\_clrspc\_mbr(clrspc))
```

Test if a color space is generic.

14.5.2.6 `jas_clrspc_isunknown`

```
#define jas_clrspc_isunknown(  
    clrspc)
```

Value:

```
((clrspc) & JAS_CLRSPC_UNKNOWNMASK)
```

Test if a color space is unknown.

14.5.2.7 jas_clrspc_mbr

```
#define jas_clrspc_mbr(  
    clrspc)
```

Value:

```
((clrspc) & 0xff)
```

Get the (family) member of a color space.

14.5.2.8 JAS_CLRSPC_UNKNOWN

```
#define JAS_CLRSPC_UNKNOWN JAS_CLRSPC_UNKNOWNMASK
```

Specific color spaces.

14.5.2.9 jas_cmprof_clrspc

```
#define jas_cmprof_clrspc(  
    prof)
```

Value:

```
((prof)->clrspc)
```

Get the color space associated with a color-management profile.

Returns

14.5.2.10 JAS_CMXFORM_NUMINTENTS

```
#define JAS_CMXFORM_NUMINTENTS 4
```

Number of rendering intents.

14.5.3 Typedef Documentation

14.5.3.1 jas_clrspc_t

```
typedef unsigned jas_clrspc_t
```

Color space.

14.5.3.2 `jas_cmpxfom_t`

```
typedef struct jas_cmpxfom_s jas_cmpxfom_t
```

Transform class.

14.5.3.3 `jas_cmreal_t`

```
typedef double jas_cmreal_t
```

Real-number type.

14.5.4 Enumeration Type Documentation

14.5.4.1 `jas_cmxform_intent_t`

```
enum jas_cmxform_intent_t
```

Rendering intents.

14.5.4.2 `jas_cmxform_op_t`

```
enum jas_cmxform_op_t
```

Transform operations.

14.5.4.3 `jas_cmxform_optm_t`

```
enum jas_cmxform_optm_t
```

Transform optimization.

14.5.5 Function Documentation

14.5.5.1 `jas_clrspc_numchans()`

```
unsigned jas_clrspc_numchans (  
    jas_clrspc_t clrspc)
```

Get the number of channels associated with a particular color space.

Returns

14.5.5.2 `jas_cmprof_copy()`

```
JAS_EXPORT jas_cmprof_t * jas_cmprof_copy (  
    const jas_cmprof_t * prof)
```

Copy a color-management profile.

This function creates a clone (i.e., copy) of a CM profile.

Returns

If successful, a pointer to the newly created CM profile is returned. Otherwise, a null pointer is returned.

14.5.5.3 `jas_cmprof_createfromclrspc()`

```
JAS_EXPORT jas_cmprof_t * jas_cmprof_createfromclrspc (  
    jas_clrspc_t clrspc)
```

Create a color-management profile from a color space.

The function creates a CM profile from a color space.

Returns

If successful, a pointer to the created CM profile is returned. Otherwise, a null pointer is returned.

14.5.5.4 `jas_cmprof_createfromiccprof()`

```
JAS_EXPORT jas_cmprof_t * jas_cmprof_createfromiccprof (  
    const jas_iccprof_t * iccprof)
```

Create a color-management profile from an ICC profile.

This function creates a CM profile from an ICC profile.

Returns

If successful, a pointer to the created CM profile is returned. Otherwise, a null pointer is returned.

14.5.5.5 `jas_cmprof_destroy()`

```
JAS_EXPORT void jas_cmprof_destroy (  
    jas_cmprof_t * prof)
```

Destroy a color-management profile.

14.5.5.6 `jas_cmxform_apply()`

```
JAS_EXPORT int jas_cmxform_apply (
    const jas_cmxform_t * xform,
    const jas_cmpixmap_t * in,
    jas_cmpixmap_t * out)
```

Apply a transform to data.

Returns

If successful, zero is returned. Otherwise, a nonzero value is returned.

14.5.5.7 `jas_cmxform_create()`

```
JAS_EXPORT jas_cmxform_t * jas_cmxform_create (
    const jas_cmprof_t * inprof,
    const jas_cmprof_t * outprof,
    const jas_cmprof_t * proofprof,
    jas_cmxform_op_t op,
    jas_cmxform_intent_t intent,
    jas_cmxform_optm_t optimize)
```

Create a transform from a CM profile.

Returns

If successful, a pointer to the created transform is returned. Otherwise, a null pointer is returned.

14.5.5.8 `jas_cmxform_destroy()`

```
JAS_EXPORT void jas_cmxform_destroy (
    jas_cmxform_t * xform)
```

Destroy a transform.

14.5.5.9 `jas_iccatrtab_dump()`

```
JAS_EXPORT void jas_iccatrtab_dump (
    const jas_iccatrtab_t * attrtab,
    FILE * out)
```

Dump an ICC profile attribute to a stream in human-readable format for debugging purposes.

14.5.5.10 jas_iccattrval_allowmodify()

```
JAS_EXPORT int jas_iccattrval_allowmodify (  
    jas_iccattrval_t ** attrval)
```

TODO/FIXME.

14.5.5.11 jas_iccattrval_clone()

```
JAS_EXPORT jas_iccattrval_t * jas_iccattrval_clone (  
    jas_iccattrval_t * attrval)
```

Create a copy of an ICC profile attribute.

14.5.5.12 jas_iccattrval_create()

```
JAS_EXPORT jas_iccattrval_t * jas_iccattrval_create (  
    jas_iccuint32_t type)
```

Create an ICC profile attribute.

14.5.5.13 jas_iccattrval_destroy()

```
JAS_EXPORT void jas_iccattrval_destroy (  
    jas_iccattrval_t * attrval)
```

Destroy an ICC profile attribute.

14.5.5.14 jas_iccprof_copy()

```
JAS_EXPORT jas_iccprof_t * jas_iccprof_copy (  
    const jas_iccprof_t * prof)
```

Create a copy of an ICC profile.

14.5.5.15 jas_iccprof_createfrombuf()

```
JAS_EXPORT jas_iccprof_t * jas_iccprof_createfrombuf (  
    const jas_uchar * buf,  
    unsigned len)
```

Create an ICC profile from a buffer in memory.

14.5.5.16 `jas_iccprof_createfromclrspc()`

```
JAS_EXPORT jas_iccprof_t * jas_iccprof_createfromclrspc (  
    unsigned clrspc)
```

Create an ICC profile from a color space.

14.5.5.17 `jas_iccprof_createfromcmprof()`

```
JAS_EXPORT jas_iccprof_t * jas_iccprof_createfromcmprof (  
    const jas_cmprof_t * prof)
```

Create a ICC profile from a CM profile.

This function creates an ICC profile from a CM profile.

Returns

If successful, a pointer to the created ICC profile is returned. Otherwise, a null pointer is returned.

14.5.5.18 `jas_iccprof_destroy()`

```
JAS_EXPORT void jas_iccprof_destroy (  
    jas_iccprof_t * prof)
```

Destroy an ICC profile.

14.5.5.19 `jas_iccprof_dump()`

```
JAS_EXPORT void jas_iccprof_dump (  
    const jas_iccprof_t * prof,  
    FILE * out)
```

Dump an ICC profile to a stream in human-readable format for debugging purposes.

14.5.5.20 `jas_iccprof_getattr()`

```
JAS_ATTRIBUTE_PURE JAS_EXPORT jas_iccattrval_t * jas_iccprof_getattr (  
    const jas_iccprof_t * prof,  
    jas_iccattrname_t name)
```

Get an attribute of an ICC profile.

14.5.5.21 jas_iccprof_gethdr()

```
JAS_EXPORT int jas_iccprof_gethdr (  
    const jas_iccprof_t * prof,  
    jas_icchdr_t * hdr)
```

Get the header for an ICC profile.

14.5.5.22 jas_iccprof_load()

```
JAS_EXPORT jas_iccprof_t * jas_iccprof_load (  
    jas_stream_t * in)
```

Read an ICC profile from a stream.

14.5.5.23 jas_iccprof_save()

```
JAS_EXPORT int jas_iccprof_save (  
    jas_iccprof_t * prof,  
    jas_stream_t * out)
```

Write an ICC profile to a stream.

14.5.5.24 jas_iccprof_setattr()

```
JAS_EXPORT int jas_iccprof_setattr (  
    jas_iccprof_t * prof,  
    jas_iccattnrname_t name,  
    jas_iccattnrval_t * val)
```

Set an attribute of an ICC profile.

14.5.5.25 jas_iccprof_sethdr()

```
JAS_EXPORT int jas_iccprof_sethdr (  
    jas_iccprof_t * prof,  
    const jas_icchdr_t * hdr)
```

Set the header for an ICC profile.

14.6 One- and Two-Dimensional Sequences

One- and Two-Dimensional Sequences.

Classes

- struct [jas_matrix_t](#)
Matrix type.
- struct [jas_seq2d_t](#)
Two-dimensional sequence type.
- struct [jas_seq_t](#)
One-dimensional sequence type.

Functions

- static JAS_ATTRIBUTE_PURE [jas_matind_t](#) [jas_matrix_numrows](#) (const [jas_matrix_t](#) *matrix)
Get the number of rows in a matrix.
- static JAS_ATTRIBUTE_PURE [jas_matind_t](#) [jas_matrix_numcols](#) (const [jas_matrix_t](#) *matrix)
Get the number of columns in a matrix.
- static JAS_ATTRIBUTE_PURE [jas_matind_t](#) [jas_matrix_size](#) (const [jas_matrix_t](#) *matrix)
Get the number of elements in a matrix.
- static JAS_ATTRIBUTE_PURE bool [jas_matrix_empty](#) (const [jas_matrix_t](#) *matrix)
Test if a matrix is empty (i.e., contains no elements).
- static JAS_ATTRIBUTE_PURE [jas_sequent_t](#) [jas_matrix_get](#) (const [jas_matrix_t](#) *matrix, [jas_matind_t](#) i, [jas_sequent_t](#) j)
Get a matrix element.
- static void [jas_matrix_set](#) ([jas_matrix_t](#) *matrix, [jas_matind_t](#) i, [jas_matind_t](#) j, [jas_sequent_t](#) v)
Set a matrix element.
- static JAS_ATTRIBUTE_PURE [jas_sequent_t](#) [jas_matrix_getv](#) (const [jas_matrix_t](#) *matrix, [jas_matind_t](#) i)
Get an element from a matrix that is known to be a row or column vector.
- static void [jas_matrix_setv](#) ([jas_matrix_t](#) *matrix, [jas_matind_t](#) i, [jas_sequent_t](#) v)
Set an element in a matrix that is known to be a row or column vector.
- static JAS_ATTRIBUTE_PURE [jas_sequent_t](#) * [jas_matrix_getref](#) (const [jas_matrix_t](#) *matrix, [jas_matind_t](#) i, [jas_sequent_t](#) j)
Get the address of an element in a matrix.
- static JAS_ATTRIBUTE_PURE [jas_sequent_t](#) * [jas_matrix_getvref](#) (const [jas_matrix_t](#) *matrix, [jas_matind_t](#) i)
Get a reference to a particular row of a 2-D sequence.
- JAS_EXPORT [jas_matrix_t](#) * [jas_matrix_create](#) ([jas_matind_t](#) numRows, [jas_matind_t](#) numcols)
Create a matrix with the specified dimensions.
- JAS_EXPORT void [jas_matrix_destroy](#) ([jas_matrix_t](#) *matrix)
Destroy a matrix.
- JAS_EXPORT int [jas_matrix_resize](#) ([jas_matrix_t](#) *matrix, [jas_matind_t](#) numRows, [jas_matind_t](#) numcols)
Resize a matrix. The previous contents of the matrix are lost.
- JAS_EXPORT int [jas_matrix_output](#) ([jas_matrix_t](#) *matrix, FILE *out)
Write a matrix to a C standard library stream.
- JAS_EXPORT int [jas_matrix_bindsub](#) ([jas_matrix_t](#) *mat0, [jas_matrix_t](#) *mat1, [jas_matind_t](#) r0, [jas_matind_t](#) c0, [jas_matind_t](#) r1, [jas_matind_t](#) c1)
Create a matrix that references part of another matrix.
- static int [jas_matrix_bindrow](#) ([jas_matrix_t](#) *mat0, [jas_matrix_t](#) *mat1, [jas_matind_t](#) r)
Create a matrix that is a reference to a row of another matrix.
- static int [jas_matrix_bindcol](#) ([jas_matrix_t](#) *mat0, [jas_matrix_t](#) *mat1, [jas_matind_t](#) c)
Create a matrix that is a reference to a column of another matrix.

- Create a matrix that is a reference to a column of another matrix.*

 - JAS_EXPORT void [jas_matrix_clip](#) ([jas_matrix_t](#) *matrix, [jas_sequent_t](#) minval, [jas_sequent_t](#) maxval)

Clip the values of matrix elements to the specified range.
- JAS_EXPORT void [jas_matrix_asl](#) ([jas_matrix_t](#) *matrix, unsigned n)

Arithmetic shift left of all elements in a matrix.
- JAS_EXPORT void [jas_matrix_asr](#) ([jas_matrix_t](#) *matrix, unsigned n)

Arithmetic shift right of all elements in a matrix.
- JAS_EXPORT void [jas_matrix_divpow2](#) ([jas_matrix_t](#) *matrix, unsigned n)

Almost-but-not-quite arithmetic shift right of all elements in a matrix.
- JAS_EXPORT void [jas_matrix_setall](#) ([jas_matrix_t](#) *matrix, [jas_sequent_t](#) val)

Set all elements of a matrix to the specified value.
- static JAS_ATTRIBUTE_PURE [size_t](#) [jas_matrix_rowstep](#) (const [jas_matrix_t](#) *matrix)

The spacing between rows of a matrix.
- static JAS_ATTRIBUTE_PURE [size_t](#) [jas_matrix_step](#) (const [jas_matrix_t](#) *matrix)

The spacing between columns of a matrix.
- JAS_EXPORT int [jas_matrix_cmp](#) ([jas_matrix_t](#) *mat0, [jas_matrix_t](#) *mat1)

Compare two matrices for equality.
- JAS_EXPORT [jas_matrix_t](#) * [jas_matrix_copy](#) ([jas_matrix_t](#) *x)

Copy a matrix.
- JAS_EXPORT [jas_matrix_t](#) * [jas_matrix_input](#) (FILE *)

Read a matrix from a C standard library stream.
- JAS_EXPORT [jas_seq2d_t](#) * [jas_seq2d_copy](#) ([jas_seq2d_t](#) *x)

Copy a 2-D sequence.
- JAS_EXPORT [jas_matrix_t](#) * [jas_seq2d_create](#) ([jas_matind_t](#) xstart, [jas_matind_t](#) ystart, [jas_matind_t](#) xend, [jas_matind_t](#) yend)

Create a 2-D sequence.
- static void [jas_seq2d_destroy](#) ([jas_seq2d_t](#) *s)

Destroy a 2-D sequence.
- static JAS_ATTRIBUTE_PURE [jas_matind_t](#) [jas_seq2d_xstart](#) (const [jas_seq2d_t](#) *s)

Get the starting x-coordinate of the sequence.
- static JAS_ATTRIBUTE_PURE [jas_matind_t](#) [jas_seq2d_ystart](#) (const [jas_seq2d_t](#) *s)

Get the starting y-coordinate of the sequence.
- static JAS_ATTRIBUTE_PURE [jas_matind_t](#) [jas_seq2d_xend](#) (const [jas_seq2d_t](#) *s)

Get the ending x-coordinate of the sequence.
- static JAS_ATTRIBUTE_PURE [jas_matind_t](#) [jas_seq2d_yend](#) (const [jas_seq2d_t](#) *s)

Get the ending y-coordinate of the sequence.
- static JAS_ATTRIBUTE_PURE [jas_sequent_t](#) * [jas_seq2d_getref](#) (const [jas_seq2d_t](#) *s, [jas_matind_t](#) x, [jas_matind_t](#) y)

Get a pointer (i.e., reference) to an element of a 2-D sequence.
- static JAS_ATTRIBUTE_PURE [jas_sequent_t](#) [jas_seq2d_get](#) (const [jas_seq2d_t](#) *s, [jas_matind_t](#) x, [jas_matind_t](#) y)

Get an element of a 2-D sequence.
- static JAS_ATTRIBUTE_PURE [size_t](#) [jas_seq2d_rowstep](#) (const [jas_seq2d_t](#) *s)

Get the stride between successive rows in the sequence.
- static JAS_ATTRIBUTE_PURE unsigned [jas_seq2d_width](#) (const [jas_seq2d_t](#) *s)

Get the number of columns in the sequence.
- static JAS_ATTRIBUTE_PURE unsigned [jas_seq2d_height](#) (const [jas_seq2d_t](#) *s)

- Get the number of rows in the sequence.*

 - static void `jas_seq2d_setshift` (`jas_seq2d_t` *s, `jas_matind_t` x, `jas_matind_t` y)

Set the shift (i.e., starting x- and y-coordinates) of the sequence.
- static JAS_ATTRIBUTE_PURE `jas_matind_t` `jas_seq2d_size` (const `jas_seq2d_t` *s)

Get the number of elements in the sequence.
- static JAS_ATTRIBUTE_PURE bool `jas_seq2d_empty` (const `jas_seq2d_t` *s)

Test if the sequence is empty (i.e., contains no elements).
- JAS_EXPORT int `jas_seq2d_bindsub` (`jas_matrix_t` *s, `jas_matrix_t` *s1, `jas_matind_t` xstart, `jas_matind_t` ystart, `jas_matind_t` xend, `jas_matind_t` yend)

Initialize a sequence to reference a subsequence of another sequence.
- static `jas_seq_t` * `jas_seq_create` (`jas_matind_t` start, `jas_matind_t` end)

Create a 1-D sequence.
- static void `jas_seq_destroy` (`jas_seq_t` *seq)

Destroy a 1-D sequence.
- static void `jas_seq_set` (`jas_seq_t` *seq, `jas_matind_t` i, `jas_sequent_t` v)

Set an element of a sequence.
- static JAS_ATTRIBUTE_PURE `jas_sequent_t` * `jas_seq_getref` (const `jas_seq_t` *seq, `jas_matind_t` i)

Get a pointer (i.e., reference) to an element of a sequence.
- static JAS_ATTRIBUTE_PURE `jas_sequent_t` `jas_seq_get` (const `jas_seq_t` *seq, `jas_matind_t` i)

Get an element of a sequence.
- static JAS_ATTRIBUTE_PURE `jas_matind_t` `jas_seq_start` (const `jas_seq_t` *seq)

Get the starting index of a sequence.
- static JAS_ATTRIBUTE_PURE `jas_matind_t` `jas_seq_end` (const `jas_seq_t` *seq)

Get the ending index of a sequence.

14.6.1 Detailed Description

One- and Two-Dimensional Sequences.

14.6.2 Function Documentation

14.6.2.1 `jas_matrix_asl()`

```
JAS_EXPORT void jas_matrix_asl (
    jas_matrix_t * matrix,
    unsigned n)
```

Arithmetic shift left of all elements in a matrix.

14.6.2.2 `jas_matrix_asr()`

```
JAS_EXPORT void jas_matrix_asr (
    jas_matrix_t * matrix,
    unsigned n)
```

Arithmetic shift right of all elements in a matrix.

14.6.2.3 `jas_matrix_bindcol()`

```
int jas_matrix_bindcol (  
    jas_matrix_t * mat0,  
    jas_matrix_t * mat1,  
    jas_matind_t c) [inline], [static]
```

Create a matrix that is a reference to a column of another matrix.

14.6.2.4 `jas_matrix_bindrow()`

```
int jas_matrix_bindrow (  
    jas_matrix_t * mat0,  
    jas_matrix_t * mat1,  
    jas_matind_t r) [inline], [static]
```

Create a matrix that is a reference to a row of another matrix.

14.6.2.5 `jas_matrix_bindsub()`

```
JAS_EXPORT int jas_matrix_bindsub (  
    jas_matrix_t * mat0,  
    jas_matrix_t * mat1,  
    jas_matind_t r0,  
    jas_matind_t c0,  
    jas_matind_t r1,  
    jas_matind_t c1)
```

Create a matrix that references part of another matrix.

14.6.2.6 `jas_matrix_clip()`

```
JAS_EXPORT void jas_matrix_clip (  
    jas_matrix_t * matrix,  
    jas_segent_t minval,  
    jas_segent_t maxval)
```

Clip the values of matrix elements to the specified range.

14.6.2.7 `jas_matrix_cmp()`

```
JAS_EXPORT int jas_matrix_cmp (  
    jas_matrix_t * mat0,  
    jas_matrix_t * mat1)
```

Compare two matrices for equality.

14.6.2.8 `jas_matrix_copy()`

```
JAS_EXPORT jas_matrix_t * jas_matrix_copy (  
    jas_matrix_t * x)
```

Copy a matrix.

14.6.2.9 `jas_matrix_create()`

```
JAS_EXPORT jas_matrix_t * jas_matrix_create (  
    jas_matind_t numrows,  
    jas_matind_t numcols)
```

Create a matrix with the specified dimensions.

14.6.2.10 `jas_matrix_destroy()`

```
JAS_EXPORT void jas_matrix_destroy (  
    jas_matrix_t * matrix)
```

Destroy a matrix.

14.6.2.11 `jas_matrix_divpow2()`

```
JAS_EXPORT void jas_matrix_divpow2 (  
    jas_matrix_t * matrix,  
    unsigned n)
```

Almost-but-not-quite arithmetic shift right of all elements in a matrix.

14.6.2.12 `jas_matrix_empty()`

```
JAS_ATTRIBUTE_PURE bool jas_matrix_empty (  
    const jas_matrix_t * matrix) [inline], [static]
```

Test if a matrix is empty (i.e., contains no elements).

14.6.2.13 `jas_matrix_get()`

```
JAS_ATTRIBUTE_PURE jas_segent_t jas_matrix_get (  
    const jas_matrix_t * matrix,  
    jas_matind_t i,  
    jas_matind_t j) [inline], [static]
```

Get a matrix element.

14.6.2.14 jas_matrix_getref()

```
JAS_ATTRIBUTE_PURE jas_segent_t * jas_matrix_getref (
    const jas_matrix_t * matrix,
    jas_matind_t i,
    jas_matind_t j) [inline], [static]
```

Get the address of an element in a matrix.

14.6.2.15 jas_matrix_getv()

```
JAS_ATTRIBUTE_PURE jas_segent_t jas_matrix_getv (
    const jas_matrix_t * matrix,
    jas_matind_t i) [inline], [static]
```

Get an element from a matrix that is known to be a row or column vector.

14.6.2.16 jas_matrix_getvref()

```
JAS_ATTRIBUTE_PURE jas_segent_t * jas_matrix_getvref (
    const jas_matrix_t * matrix,
    jas_matind_t i) [inline], [static]
```

Get a reference to a particular row of a 2-D sequence.

14.6.2.17 jas_matrix_input()

```
JAS_EXPORT jas_matrix_t * jas_matrix_input (
    FILE * )
```

Read a matrix from a C standard library stream.

14.6.2.18 jas_matrix_numcols()

```
JAS_ATTRIBUTE_PURE jas_matind_t jas_matrix_numcols (
    const jas_matrix_t * matrix) [inline], [static]
```

Get the number of columns in a matrix.

14.6.2.19 jas_matrix_numrows()

```
JAS_ATTRIBUTE_PURE jas_matind_t jas_matrix_numrows (
    const jas_matrix_t * matrix) [inline], [static]
```

Get the number of rows in a matrix.

14.6.2.20 `jas_matrix_output()`

```
JAS_EXPORT int jas_matrix_output (
    jas_matrix_t * matrix,
    FILE * out)
```

Write a matrix to a C standard library stream.

14.6.2.21 `jas_matrix_resize()`

```
JAS_EXPORT int jas_matrix_resize (
    jas_matrix_t * matrix,
    jas_matind_t numRows,
    jas_matind_t numcols)
```

Resize a matrix. The previous contents of the matrix are lost.

14.6.2.22 `jas_matrix_rowstep()`

```
JAS_ATTRIBUTE_PURE size_t jas_matrix_rowstep (
    const jas_matrix_t * matrix) [inline], [static]
```

The spacing between rows of a matrix.

14.6.2.23 `jas_matrix_set()`

```
void jas_matrix_set (
    jas_matrix_t * matrix,
    jas_matind_t i,
    jas_matind_t j,
    jas_segent_t v) [inline], [static]
```

Set a matrix element.

14.6.2.24 `jas_matrix_setall()`

```
JAS_EXPORT void jas_matrix_setall (
    jas_matrix_t * matrix,
    jas_segent_t val)
```

Set all elements of a matrix to the specified value.

14.6.2.25 jas_matrix_setv()

```
void jas_matrix_setv (  
    jas_matrix_t * matrix,  
    jas_matind_t i,  
    jas_seqent_t v) [inline], [static]
```

Set an element in a matrix that is known to be a row or column vector.

14.6.2.26 jas_matrix_size()

```
JAS_ATTRIBUTE_PURE jas_matind_t jas_matrix_size (  
    const jas_matrix_t * matrix) [inline], [static]
```

Get the number of elements in a matrix.

14.6.2.27 jas_matrix_step()

```
JAS_ATTRIBUTE_PURE size_t jas_matrix_step (  
    const jas_matrix_t * matrix) [inline], [static]
```

The spacing between columns of a matrix.

14.6.2.28 jas_seq2d_bindsub()

```
JAS_EXPORT int jas_seq2d_bindsub (  
    jas_matrix_t * s,  
    jas_matrix_t * sl,  
    jas_matind_t xstart,  
    jas_matind_t ystart,  
    jas_matind_t xend,  
    jas_matind_t yend)
```

Initialize a sequence to reference a subsequence of another sequence.

14.6.2.29 jas_seq2d_copy()

```
JAS_EXPORT jas_seq2d_t * jas_seq2d_copy (  
    jas_seq2d_t * x)
```

Copy a 2-D sequence.

14.6.2.30 `jas_seq2d_create()`

```
JAS_EXPORT jas_matrix_t * jas_seq2d_create (  
    jas_matind_t xstart,  
    jas_matind_t ystart,  
    jas_matind_t xend,  
    jas_matind_t yend)
```

Create a 2-D sequence.

14.6.2.31 `jas_seq2d_destroy()`

```
void jas_seq2d_destroy (  
    jas_seq2d_t * s) [inline], [static]
```

Destroy a 2-D sequence.

14.6.2.32 `jas_seq2d_empty()`

```
JAS_ATTRIBUTE_PURE bool jas_seq2d_empty (  
    const jas_seq2d_t * s) [inline], [static]
```

Test if the sequence is empty (i.e., contains no elements).

14.6.2.33 `jas_seq2d_get()`

```
JAS_ATTRIBUTE_PURE jas_seqent_t jas_seq2d_get (  
    const jas_seq2d_t * s,  
    jas_matind_t x,  
    jas_matind_t y) [inline], [static]
```

Get an element of a 2-D sequence.

14.6.2.34 `jas_seq2d_getref()`

```
JAS_ATTRIBUTE_PURE jas_seqent_t * jas_seq2d_getref (  
    const jas_seq2d_t * s,  
    jas_matind_t x,  
    jas_matind_t y) [inline], [static]
```

Get a pointer (i.e., reference) to an element of a 2-D sequence.

14.6.2.35 `jas_seq2d_height()`

```
JAS_ATTRIBUTE_PURE unsigned jas_seq2d_height (  
    const jas_seq2d_t * s) [inline], [static]
```

Get the number of rows in the sequence.

14.6.2.36 jas_seq2d_rowstep()

```
JAS_ATTRIBUTE_PURE size_t jas_seq2d_rowstep (  
    const jas_seq2d_t * s) [inline], [static]
```

Get the stride between successive rows in the sequence.

14.6.2.37 jas_seq2d_setshift()

```
void jas_seq2d_setshift (  
    jas_seq2d_t * s,  
    jas_matind_t x,  
    jas_matind_t y) [inline], [static]
```

Set the shift (i.e., starting x- and y-coordinates) of the sequence.

14.6.2.38 jas_seq2d_size()

```
JAS_ATTRIBUTE_PURE jas_matind_t jas_seq2d_size (  
    const jas_seq2d_t * s) [inline], [static]
```

Get the number of elements in the sequence.

14.6.2.39 jas_seq2d_width()

```
JAS_ATTRIBUTE_PURE unsigned jas_seq2d_width (  
    const jas_seq2d_t * s) [inline], [static]
```

Get the number of columns in the sequence.

14.6.2.40 jas_seq2d_xend()

```
JAS_ATTRIBUTE_PURE jas_matind_t jas_seq2d_xend (  
    const jas_seq2d_t * s) [inline], [static]
```

Get the ending x-coordinate of the sequence.

14.6.2.41 jas_seq2d_xstart()

```
JAS_ATTRIBUTE_PURE jas_matind_t jas_seq2d_xstart (  
    const jas_seq2d_t * s) [inline], [static]
```

Get the starting x-coordinate of the sequence.

14.6.2.42 jas_seq2d_yend()

```
JAS_ATTRIBUTE_PURE jas_matind_t jas_seq2d_yend (  
    const jas_seq2d_t * s) [inline], [static]
```

Get the ending y-coordinate of the sequence.

14.6.2.43 jas_seq2d_ystart()

```
JAS_ATTRIBUTE_PURE jas_matind_t jas_seq2d_ystart (  
    const jas_seq2d_t * s) [inline], [static]
```

Get the starting y-coordinate of the sequence.

14.6.2.44 jas_seq_create()

```
jas_seq_t * jas_seq_create (  
    jas_matind_t start,  
    jas_matind_t end) [inline], [static]
```

Create a 1-D sequence.

14.6.2.45 jas_seq_destroy()

```
void jas_seq_destroy (  
    jas_seq_t * seq) [inline], [static]
```

Destroy a 1-D sequence.

14.6.2.46 jas_seq_end()

```
JAS_ATTRIBUTE_PURE jas_matind_t jas_seq_end (  
    const jas_seq_t * seq) [inline], [static]
```

Get the ending index of a sequence.

14.6.2.47 jas_seq_get()

```
JAS_ATTRIBUTE_PURE jas_seqent_t jas_seq_get (  
    const jas_seq_t * seq,  
    jas_matind_t i) [inline], [static]
```

Get an element of a sequence.

14.6.2.48 jas_seq_getref()

```
JAS_ATTRIBUTE_PURE jas_seqent_t * jas_seq_getref (
    const jas_seq_t * seq,
    jas_matind_t i) [inline], [static]
```

Get a pointer (i.e., reference) to an element of a sequence.

14.6.2.49 jas_seq_set()

```
void jas_seq_set (
    jas_seq_t * seq,
    jas_matind_t i,
    jas_seqent_t v) [inline], [static]
```

Set an element of a sequence.

14.6.2.50 jas_seq_start()

```
JAS_ATTRIBUTE_PURE jas_matind_t jas_seq_start (
    const jas_seq_t * seq) [inline], [static]
```

Get the starting index of a sequence.

14.7 Fixed-Point Arithmetic

Fixed-Point Arithmetic.

Macros

- #define JAS_FIX_ZERO(fix_t, fracbits)
- #define JAS_FIX_ONE(fix_t, fracbits)
- #define JAS_FIX_HALF(fix_t, fracbits)
- #define JAS_INTTOFIX(fix_t, fracbits, x)
- #define JAS_FIXTOINT(fix_t, fracbits, x)
- #define JAS_FIXTODBL(fix_t, fracbits, x)
- #define JAS_DBLTOFIX(fix_t, fracbits, x)
- #define JAS_FIX_ADD JAS_FIX_ADD_FAST
- #define JAS_FIX_ADD_FAST(fix_t, fracbits, x, y)
- #define JAS_FIX_ADD_OFLOW(fix_t, fracbits, x, y)
- #define JAS_FIX_MUL JAS_FIX_MUL_FAST
- #define JAS_FIX_MUL_FAST(fix_t, fracbits, bigfix_t, x, y)
- #define JAS_FIX_MUL_OFLOW(fix_t, fracbits, bigfix_t, x, y)
- #define JAS_FIX_MULBYINT JAS_FIX_MULBYINT_FAST
- #define JAS_FIX_MULBYINT_FAST(fix_t, fracbits, x, y)

- `#define JAS_FIX_MULBYINT_OFLOW(fix_t, fracbits, x, y)`
- `#define JAS_FIX_DIV JAS_FIX_DIV_FAST`
- `#define JAS_FIX_DIV_FAST(fix_t, fracbits, bigfix_t, x, y)`
- `#define JAS_FIX_DIV_UFLOW(fix_t, fracbits, bigfix_t, x, y)`
- `#define JAS_FIX_NEG JAS_FIX_NEG_FAST`
- `#define JAS_FIX_NEG_FAST(fix_t, fracbits, x)`
- `#define JAS_FIX_NEG_OFLOW(fix_t, fracbits, x)`
- `#define JAS_FIX_ASF JAS_FIX_ASF_FAST`
- `#define JAS_FIX_ASF_FAST(fix_t, fracbits, x, n)`
- `#define JAS_FIX_ASF_UFLOW(fix_t, fracbits, x, n)`
- `#define JAS_FIX_ASR JAS_FIX_ASR_FAST`
- `#define JAS_FIX_ASR_FAST(fix_t, fracbits, x, n)`
- `#define JAS_FIX_ASR_UFLOW(fix_t, fracbits, x, n)`
- `#define JAS_FIX_SUB(fix_t, fracbits, x, y)`
- `#define JAS_FIX_PLUSEQ(fix_t, fracbits, x, y)`
- `#define JAS_FIX_MINUSEQ(fix_t, fracbits, x, y)`
- `#define JAS_FIX_MULEQ(fix_t, fracbits, bigfix_t, x, y)`
- `#define JAS_FIX_ABS(fix_t, fracbits, x)`
- `#define JAS_FIX_ISINT(fix_t, fracbits, x)`
- `#define JAS_FIX_SGN(fix_t, fracbits, x)`
- `#define JAS_FIX_CMP(fix_t, fracbits, x, y)`
- `#define JAS_FIX_LT(fix_t, fracbits, x, y)`
- `#define JAS_FIX_LTE(fix_t, fracbits, x, y)`
- `#define JAS_FIX_GT(fix_t, fracbits, x, y)`
- `#define JAS_FIX_GTE(fix_t, fracbits, x, y)`
- `#define JAS_FIX_ROUND(fix_t, fracbits, x)`
- `#define JAS_FIX_FLOOR(fix_t, fracbits, x)`

Typedefs

- `typedef int_least64_t jas_fix_t`

14.7.1 Detailed Description

Fixed-Point Arithmetic.

14.7.2 Macro Definition Documentation

14.7.2.1 JAS_DBLTOFIX

```
#define JAS_DBLTOFIX(  
    fix_t,  
    fracbits,  
    x)
```

Value:

```
JAS_CAST(fix_t, ((x) * JAS_CAST(double, JAS_FIX_ONE(fix_t, fracbits))))
```

Convert a double to a fixed-point number.

14.7.2.2 JAS_FIX_ABS

```
#define JAS_FIX_ABS(  
    fix_t,  
    fracbits,  
    x)
```

Value:

```
((x) >= 0) ? (x) : (JAS_FIX_NEG(fix_t, fracbits, x))
```

Calculate the absolute value of a fixed-point number.

14.7.2.3 JAS_FIX_ADD

```
#define JAS_FIX_ADD JAS_FIX_ADD_FAST
```

Calculate the sum of two fixed-point numbers.

14.7.2.4 JAS_FIX_ADD_FAST

```
#define JAS_FIX_ADD_FAST(  
    fix_t,  
    fracbits,  
    x,  
    y)
```

Value:

```
((x) + (y))
```

Calculate the sum of two fixed-point numbers without overflow checking.

14.7.2.5 JAS_FIX_ADD_OFLOW

```
#define JAS_FIX_ADD_OFLOW(  
    fix_t,  
    fracbits,  
    x,  
    y)
```

Value:

```
((x) >= 0) ? \  
    ((y) >= 0) ? ((x) + (y) >= 0 || JAS_FIX_OFLOW(), (x) + (y)) : \  
    ((x) + (y)) : \  
    ((y) < 0) ? ((x) + (y)) : ((x) + (y) < 0 || JAS_FIX_OFLOW(), \  
    (x) + (y))
```

Calculate the sum of two fixed-point numbers with overflow checking.

14.7.2.6 JAS_FIX_ASL

```
#define JAS_FIX_ASL JAS_FIX_ASL_FAST
```

Perform an arithmetic shift left of a fixed-point number.

14.7.2.7 JAS_FIX_ASL_FAST

```
#define JAS_FIX_ASL_FAST(  
    fix_t,  
    fracbits,  
    x,  
    n)
```

Value:

```
((x) << (n))
```

Perform an arithmetic shift left of a fixed-point number without overflow checking.

14.7.2.8 JAS_FIX_ASL_OFLOW

```
#define JAS_FIX_ASL_OFLOW(  
    fix_t,  
    fracbits,  
    x,  
    n)
```

Value:

```
((((x) << (n)) >> (n)) == (x) || JAS_FIX_OFLOW(), (x) << (n))
```

Perform an arithmetic shift left of a fixed-point number with overflow checking.

14.7.2.9 JAS_FIX_ASR

```
#define JAS_FIX_ASR JAS_FIX_ASR_FAST
```

Perform an arithmetic shift right of a fixed-point number.

14.7.2.10 JAS_FIX_ASR_FAST

```
#define JAS_FIX_ASR_FAST(  
    fix_t,  
    fracbits,  
    x,  
    n)
```

Value:

```
((x) >> (n))
```

Perform an arithmetic shift right of a fixed-point number without underflow checking.

14.7.2.11 JAS_FIX_ASR_UFLOW

```
#define JAS_FIX_ASR_UFLOW(  
    fix_t,  
    fracbits,  
    x,  
    n)
```

Value:

`JAS_FIX_ASR_FAST(fix_t, fracbits, x, n)`

Perform an arithmetic shift right of a fixed-point number with underflow checking.

14.7.2.12 JAS_FIX_CMP

```
#define JAS_FIX_CMP(  
    fix_t,  
    fracbits,  
    x,  
    y)
```

Value:

`((x) > (y) ? 1 : (((x) == (y)) ? 0 : (-1)))`

Compare two fixed-point numbers.

14.7.2.13 JAS_FIX_DIV

```
#define JAS_FIX_DIV JAS_FIX_DIV_FAST
```

Calculate the quotient of two fixed-point numbers.

14.7.2.14 JAS_FIX_DIV_FAST

```
#define JAS_FIX_DIV_FAST(  
    fix_t,  
    fracbits,  
    bigfix_t,  
    x,  
    y)
```

Value:

`JAS_CAST(fix_t, (JAS_CAST(bigfix_t, x) « (fracbits)) / (y))`

Calculate the quotient of two fixed-point numbers without underflow checking.

14.7.2.15 JAS_FIX_DIV_UFLOW

```
#define JAS_FIX_DIV_UFLOW(  
    fix_t,  
    fracbits,  
    bigfix_t,  
    x,  
    y)
```

Value:

`JAS_FIX_DIV_FAST(fix_t, fracbits, bigfix_t, x, y)`

Calculate the quotient of two fixed-point numbers with underflow checking.

14.7.2.16 JAS_FIX_FLOOR

```
#define JAS_FIX_FLOOR(  
    fix_t,  
    fracbits,  
    x)
```

Value:

`((x) & ~(JAS_FIX_ONE(fix_t, fracbits) - 1))`

Round a fixed-point number to the nearest integer in the direction of negative infinity (i.e., the floor function).

14.7.2.17 JAS_FIX_GT

```
#define JAS_FIX_GT(  
    fix_t,  
    fracbits,  
    x,  
    y)
```

Value:

`((x) > (y))`

Greater than.

14.7.2.18 JAS_FIX_GTE

```
#define JAS_FIX_GTE(  
    fix_t,  
    fracbits,  
    x,  
    y)
```

Value:

`((x) >= (y))`

Greater than or equal.

14.7.2.19 JAS_FIX_HALF

```
#define JAS_FIX_HALF(  
    fix_t,  
    fracbits)
```

Value:

```
(JAS_CAST(fix_t, 1) « ((fracbits) - 1))
```

The representation of the value one half.

14.7.2.20 JAS_FIX_ISINT

```
#define JAS_FIX_ISINT(  
    fix_t,  
    fracbits,  
    x)
```

Value:

```
(JAS_FIX_FLOOR(fix_t, fracbits, x) == (x))
```

Is a fixed-point number an integer?

14.7.2.21 JAS_FIX_LT

```
#define JAS_FIX_LT(  
    fix_t,  
    fracbits,  
    x,  
    y)
```

Value:

```
((x) < (y))
```

Less than.

14.7.2.22 JAS_FIX_LTE

```
#define JAS_FIX_LTE(  
    fix_t,  
    fracbits,  
    x,  
    y)
```

Value:

```
((x) <= (y))
```

Less than or equal.

14.7.2.23 JAS_FIX_MINUSEQ

```
#define JAS_FIX_MINUSEQ(  
    fix_t,  
    fracbits,  
    x,  
    y)
```

Value:

```
((x) = JAS_FIX_SUB(fix_t, fracbits, x, y))
```

Subtract one fixed-point number from another.

14.7.2.24 JAS_FIX_MUL

```
#define JAS_FIX_MUL JAS_FIX_MUL_FAST
```

Calculate the product of two fixed-point numbers.

14.7.2.25 JAS_FIX_MUL_FAST

```
#define JAS_FIX_MUL_FAST(  
    fix_t,  
    fracbits,  
    bigfix_t,  
    x,  
    y)
```

Value:

```
JAS_CAST(fix_t, (JAS_CAST(bigfix_t, x) * JAS_CAST(bigfix_t, y)) » \  
    (fracbits))
```

Calculate the product of two fixed-point numbers without overflow checking.

14.7.2.26 JAS_FIX_MUL_OFLOW

```
#define JAS_FIX_MUL_OFLOW(  
    fix_t,  
    fracbits,  
    bigfix_t,  
    x,  
    y)
```

Value:

```
((JAS_CAST(bigfix_t, x) * JAS_CAST(bigfix_t, y)) » (fracbits)) == \  
    JAS_CAST(fix_t, (JAS_CAST(bigfix_t, x) * JAS_CAST(bigfix_t, y)) » \  
    (fracbits))) ? \  
    JAS_CAST(fix_t, (JAS_CAST(bigfix_t, x) * JAS_CAST(bigfix_t, y)) » \  
    (fracbits))) : JAS_FIX_OFLOW()
```

Calculate the product of two fixed-point numbers with overflow checking.

14.7.2.27 JAS_FIX_MULBYINT

```
#define JAS_FIX_MULBYINT JAS_FIX_MULBYINT_FAST
```

Calculate the product of a fixed-point number and an int.

14.7.2.28 JAS_FIX_MULBYINT_FAST

```
#define JAS_FIX_MULBYINT_FAST(  
    fix_t,  
    fracbits,  
    x,  
    y)
```

Value:

```
JAS_CAST(fix_t, ((x) * (y)))
```

Calculate the product of a fixed-point number and an int without overflow checking.

14.7.2.29 JAS_FIX_MULBYINT_OFLOW

```
#define JAS_FIX_MULBYINT_OFLOW(  
    fix_t,  
    fracbits,  
    x,  
    y)
```

Value:

```
JAS_FIX_MULBYINT_FAST(fix_t, fracbits, x, y)
```

Calculate the product of a fixed-point number and an int with overflow checking.

14.7.2.30 JAS_FIX_MULEQ

```
#define JAS_FIX_MULEQ(  
    fix_t,  
    fracbits,  
    bigfix_t,  
    x,  
    y)
```

Value:

```
((x) = JAS_FIX_MUL(fix_t, fracbits, bigfix_t, x, y))
```

Multiply one fixed-point number by another.

14.7.2.31 JAS_FIX_NEG

```
#define JAS_FIX_NEG JAS_FIX_NEG_FAST
```

Negate a fixed-point number.

14.7.2.32 JAS_FIX_NEG_FAST

```
#define JAS_FIX_NEG_FAST(  
    fix_t,  
    fracbits,  
    x)
```

Value:

```
(-(x))
```

Negate a fixed-point number without overflow checking.

14.7.2.33 JAS_FIX_NEG_OFLOW

```
#define JAS_FIX_NEG_OFLOW(  
    fix_t,  
    fracbits,  
    x)
```

Value:

```
((x) < 0) ? (-(x) > 0 || JAS_FIX_OFLOW(), -(x)) : (-(x))
```

Negate a fixed-point number with overflow checking.

14.7.2.34 JAS_FIX_ONE

```
#define JAS_FIX_ONE(  
    fix_t,  
    fracbits)
```

Value:

```
JAS_INTTOFIX(fix_t, fracbits, 1)
```

The representation of the value one.

14.7.2.35 JAS_FIX_PLUSEQ

```
#define JAS_FIX_PLUSEQ(  
    fix_t,  
    fracbits,  
    x,  
    y)
```

Value:

```
((x) = JAS_FIX_ADD(fix_t, fracbits, x, y))
```

Add one fixed-point number to another.

14.7.2.36 JAS_FIX_ROUND

```
#define JAS_FIX_ROUND(  
    fix_t,  
    fracbits,  
    x)
```

Value:

```
((x) < 0) ? JAS_FIX_FLOOR(fix_t, fracbits, JAS_FIX_ADD(fix_t, fracbits, \  
    (x), JAS_FIX_HALF(fix_t, fracbits))) : \  
    JAS_FIX_NEG(fix_t, fracbits, JAS_FIX_FLOOR(fix_t, fracbits, \  
    JAS_FIX_ADD(fix_t, fracbits, (-(x)), JAS_FIX_HALF(fix_t, fracbits))))
```

Round a fixed-point number to the nearest integer.

14.7.2.37 JAS_FIX_SGN

```
#define JAS_FIX_SGN(  
    fix_t,  
    fracbits,  
    x)
```

Value:

```
((x) >= 0 ? 1 : (-1))
```

Get the sign of a fixed-point number.

14.7.2.38 JAS_FIX_SUB

```
#define JAS_FIX_SUB(  
    fix_t,  
    fracbits,  
    x,  
    y)
```

Value:

```
JAS_FIX_ADD(fix_t, fracbits, x, JAS_FIX_NEG(fix_t, fracbits, y))
```

Calculate the difference between two fixed-point numbers.

14.7.2.39 JAS_FIX_ZERO

```
#define JAS_FIX_ZERO(  
    fix_t,  
    fracbits)
```

Value:

```
JAS_INTTOFIX(fix_t, fracbits, 0)
```

The representation of the value zero.

14.7.2.40 JAS_FIXTODBL

```
#define JAS_FIXTODBL(  
    fix_t,  
    fracbits,  
    x)
```

Value:

```
(JAS_CAST(double, x) / JAS_FIX_ONE(fix_t, fracbits))
```

Convert a fixed-point number to a double.

14.7.2.41 JAS_FIXTOINT

```
#define JAS_FIXTOINT(  
    fix_t,  
    fracbits,  
    x)
```

Value:

```
JAS_CAST(int, (x) » (fracbits))
```

Convert a fixed-point number to an int.

14.7.2.42 JAS_INTTOFIX

```
#define JAS_INTTOFIX(  
    fix_t,  
    fracbits,  
    x)
```

Value:

```
(JAS_CAST(fix_t, x) « (fracbits))
```

Convert an int to a fixed-point number.

14.7.3 Typedef Documentation

14.7.3.1 jas_fix_t

```
typedef int_least64_t jas_fix_t
```

Integral type used for fixed-point representations.

14.8 Logging

Logging.

Classes

- struct `jas_logtype_t`
Type used for the log type.

Macros

- `#define JAS_LOGTYPE_CLASS_NULL 0`
- `#define JAS_LOGTYPE_CLASS_ERROR 1`
- `#define JAS_LOGTYPE_CLASS_WARN 2`
- `#define JAS_LOGTYPE_CLASS_INFO 3`
- `#define JAS_LOGTYPE_CLASS_DEBUG 4`

Typedefs

- typedef int `jas_vlogmsgf_t(jas_logtype_t, const char *, va_list)`
Type used for formatted message logging function.

Functions

- static `jas_logtype_t jas_logtype_init` (int clas, int priority)
Create an instance of a logtype.
- static int `jas_logtype_getclass` (jas_logtype_t type)
Get the class of a logtype.
- static int `jas_logtype_getpriority` (jas_logtype_t type)
Get the priority of a logtype.
- JAS_EXPORT int `jas_vlogmsgf` (jas_logtype_t type, const char *fmt, va_list ap)
Print formatted log message.
- JAS_EXPORT int `jas_vlogmsgf_stderr` (jas_logtype_t type, const char *fmt, va_list ap)
Output a log message to standard error.
- JAS_EXPORT int `jas_vlogmsgf_discard` (jas_logtype_t type, const char *fmt, va_list ap)
Output a log message to nowhere (i.e., discard the message).

14.8.1 Detailed Description

Logging.

General information can be found [here](#).

14.8.2 Macro Definition Documentation

14.8.2.1 JAS_LOGTYPE_CLASS_DEBUG

```
#define JAS_LOGTYPE_CLASS_DEBUG 4
```

Log type class for debugging messages.

14.8.2.2 JAS_LOGTYPE_CLASS_ERROR

```
#define JAS_LOGTYPE_CLASS_ERROR 1
```

Log type class for errors.

14.8.2.3 JAS_LOGTYPE_CLASS_INFO

```
#define JAS_LOGTYPE_CLASS_INFO 3
```

Log type class for informational messages.

14.8.2.4 JAS_LOGTYPE_CLASS_NULL

```
#define JAS_LOGTYPE_CLASS_NULL 0
```

Log type class for unclassified messages.

14.8.2.5 JAS_LOGTYPE_CLASS_WARN

```
#define JAS_LOGTYPE_CLASS_WARN 2
```

Log type class for warnings.

14.8.3 Typedef Documentation

14.8.3.1 jas_vlogmsgf_t

```
typedef int jas_vlogmsgf_t(jas_logtype_t, const char *, va_list)
```

Type used for formatted message logging function.

14.8.4 Function Documentation

14.8.4.1 jas_logtype_getclass()

```
int jas_logtype_getclass (  
    jas_logtype_t type) [inline], [static]
```

Get the class of a logtype.

14.8.4.2 `jas_logtype_getpriority()`

```
int jas_logtype_getpriority (  
    jas_logtype_t type) [inline], [static]
```

Get the priority of a logtype.

14.8.4.3 `jas_logtype_init()`

```
jas_logtype_t jas_logtype_init (  
    int clas,  
    int priority) [inline], [static]
```

Create an instance of a logtype.

14.8.4.4 `jas_vlogmsgf()`

```
JAS_EXPORT int jas_vlogmsgf (  
    jas_logtype_t type,  
    const char * fmt,  
    va_list ap)
```

Print formatted log message.

14.8.4.5 `jas_vlogmsgf_discard()`

```
JAS_EXPORT int jas_vlogmsgf_discard (  
    jas_logtype_t type,  
    const char * fmt,  
    va_list ap)
```

Output a log message to nowhere (i.e., discard the message).

14.8.4.6 `jas_vlogmsgf_stderr()`

```
JAS_EXPORT int jas_vlogmsgf_stderr (  
    jas_logtype_t type,  
    const char * fmt,  
    va_list ap)
```

Output a log message to standard error.

14.9 Timers

Timers.

Classes

- struct `jas_tmr_t`
Timer type.

Functions

- JAS_EXPORT void `jas_tmr_start` (`jas_tmr_t *tmr`)
Start a timer.
- JAS_EXPORT void `jas_tmr_stop` (`jas_tmr_t *tmr`)
Stop a timer.
- JAS_EXPORT double `jas_tmr_get` (`jas_tmr_t *tmr`)
Get the elapsed time for a timer.

14.9.1 Detailed Description

Timers.

General information can be found [here](#).

14.9.2 Function Documentation

14.9.2.1 `jas_tmr_get()`

```
JAS_EXPORT double jas_tmr_get (  
    jas_tmr_t * tmr)
```

Get the elapsed time for a timer.

14.9.2.2 `jas_tmr_start()`

```
JAS_EXPORT void jas_tmr_start (  
    jas_tmr_t * tmr)
```

Start a timer.

14.9.2.3 `jas_tmr_stop()`

```
JAS_EXPORT void jas_tmr_stop (  
    jas_tmr_t * tmr)
```

Stop a timer.

14.10 Command-Line Interface (CLI) Option Processing

Command-Line Interface (CLI) Option Processing.

Classes

- struct `jas_opt_t`
Command line option type.

Macros

- #define `JAS_GETOPT_EOF` (-1)
- #define `JAS_GETOPT_ERR` '?'
- #define `JAS_OPT_HASARG` 0x01 /* option has argument */

Functions

- JAS_EXPORT int `jas_getopt` (int argc, char **argv, const `jas_opt_t` *opts)
Get the next option.

Variables

- JAS_EXPORT int `jas_optind`
The current option index.
- JAS_EXPORT const char * `jas_optarg`
The current option argument.
- JAS_EXPORT int `jas_opterr`
The debug level.

14.10.1 Detailed Description

Command-Line Interface (CLI) Option Processing.

General information can be found [here](#).

14.10.2 Macro Definition Documentation

14.10.2.1 JAS_GETOPT_EOF

```
#define JAS_GETOPT_EOF (-1)
```

Last CLI option.

14.10.2.2 JAS_GETOPT_ERR

```
#define JAS_GETOPT_ERR '?'
```

Error while processing CLI options.

14.10.2.3 JAS_OPT_HASARG

```
#define JAS_OPT_HASARG 0x01 /* option has argument */
```

Option has argument.

14.10.3 Function Documentation

14.10.3.1 jas_getopt()

```
JAS_EXPORT int jas_getopt (  
    int argc,  
    char ** argv,  
    const jas_opt_t * opts)
```

Get the next option.

Gets the next CLI option.

Warning

This function is not thread safe, due to its use of `jas_optind`, `jas_optarg`, and `jas_opterr`.

14.10.4 Variable Documentation

14.10.4.1 jas_optarg

```
JAS_EXPORT const char* jas_optarg [extern]
```

The current option argument.

14.10.4.2 jas_opterr

```
JAS_EXPORT int jas_opterr [extern]
```

The debug level.

14.10.4.3 jas_optind

```
JAS_EXPORT int jas_optind [extern]
```

The current option index.

14.11 Tag-Value Pair (TVP) Parsing

Tag-Value Pair (TVP) Parsing.

Classes

- struct [jas_taginfo_t](#)
Tag information type.
- struct [jas_tvparser_t](#)
Tag-value parser type.

Functions

- JAS_ATTRIBUTE_PURE JAS_EXPORT const [jas_taginfo_t](#) * [jas_taginfos_lookup](#) (const [jas_taginfo_t](#) *taginfos, const char *name)
Lookup a tag by name.
- JAS_ATTRIBUTE_PURE JAS_EXPORT const [jas_taginfo_t](#) * [jas_taginfo_nonnull](#) (const [jas_taginfo_t](#) *taginfo)
Ensure a nonnull taginfo pointer.
- JAS_EXPORT [jas_tvparser_t](#) * [jas_tvparser_create](#) (const char *s)
Create a tag-value parser for the specified string.
- JAS_EXPORT void [jas_tvparser_destroy](#) ([jas_tvparser_t](#) *tvparser)
Destroy a tag-value parser.
- JAS_EXPORT int [jas_tvparser_next](#) ([jas_tvparser_t](#) *tvparser)
Get the next tag-value pair.
- JAS_ATTRIBUTE_PURE JAS_EXPORT const char * [jas_tvparser_gettag](#) (const [jas_tvparser_t](#) *tvparser)
Get the tag name for the current tag-value pair.
- JAS_ATTRIBUTE_PURE JAS_EXPORT const char * [jas_tvparser_getval](#) (const [jas_tvparser_t](#) *tvparser)
Get the value for the current tag-value pair.

14.11.1 Detailed Description

Tag-Value Pair (TVP) Parsing.

General information can be found [here](#).

14.11.2 Function Documentation

14.11.2.1 `jas_taginfo_nonull()`

```
JAS_ATTRIBUTE_PURE JAS_EXPORT const jas\_taginfo\_t * jas_taginfo_nonull (  
    const jas\_taginfo\_t * taginfo)
```

Ensure a nonnull taginfo pointer.

This function returns a pointer to the specified taginfo object if it exists (i.e., the pointer is nonnull); otherwise, a pointer to a dummy object is returned. This is useful in some situations to avoid checking for a null pointer.

14.11.2.2 `jas_taginfos_lookup()`

```
JAS_ATTRIBUTE_PURE JAS_EXPORT const jas\_taginfo\_t * jas_taginfos_lookup (  
    const jas\_taginfo\_t * taginfos,  
    const char * name)
```

Lookup a tag by name.

14.11.2.3 `jas_tvparser_create()`

```
JAS_EXPORT jas\_tvparser\_t * jas_tvparser_create (  
    const char * s)
```

Create a tag-value parser for the specified string.

14.11.2.4 `jas_tvparser_destroy()`

```
JAS_EXPORT void jas_tvparser_destroy (  
    jas\_tvparser\_t * tvparser)
```

Destroy a tag-value parser.

14.11.2.5 `jas_tvparser_gettag()`

```
JAS_ATTRIBUTE_PURE JAS_EXPORT const char * jas_tvparser_gettag (  
    const jas\_tvparser\_t * tvparser)
```

Get the tag name for the current tag-value pair.

14.11.2.6 `jas_tvparser_getval()`

```
JAS_ATTRIBUTE_PURE JAS_EXPORT const char * jas_tvparser_getval (  
    const jas\_tvparser\_t * tvparser)
```

Get the value for the current tag-value pair.

14.11.2.7 jas_tvparser_next()

```
JAS_EXPORT int jas_tvparser_next (
    jas_tvparser_t * tvparser)
```

Get the next tag-value pair.

14.12 String Processing

String Processing.

Functions

- JAS_EXPORT char * [jas_strdup](#) (const char *)
Create a copy of a null-terminated string.
- JAS_EXPORT char * [jas_strtok](#) (char *str, const char *delim, char **saveptr)
Extract tokens from a string.
- JAS_EXPORT int [jas_stringtokenize](#) (const char *string, const char *delim, char ***tokens_buf, size_t *max_tokens_buf, size_t *num_tokens_buf)
Split a string into tokens based on specified delimiters.

14.12.1 Detailed Description

String Processing.

General information can be found [here](#).

14.12.2 Function Documentation

14.12.2.1 jas_strdup()

```
JAS_EXPORT char * jas_strdup (
    const char * s)
```

Create a copy of a null-terminated string.

This function has a behavior similar to the well-known strdup function.

14.12.2.2 jas_stringtokenize()

```
JAS_EXPORT int jas_stringtokenize (
    const char * string,
    const char * delim,
    char *** tokens_buf,
    size_t * max_tokens_buf,
    size_t * num_tokens_buf)
```

Parameters

<i>string</i>	A pointer to a null-terminated string to be split into tokens.
<i>delim</i>	A pointer to a null-terminated string contained characters used to delimit tokens.
<i>tokens_buf</i>	A pointer to the output token array.
<i>max_tokens_buf</i>	A pointer to the allocated size of the token array.
<i>num_tokens_buf</i>	A pointer to the number of elements in the token array.

The memory to hold token information is allocated via [jas_malloc\(\)](#) and friends.

Returns

If successful, zero is returned. Otherwise, a nonzero value is returned.

14.12.2.3 `jas_strtok()`

```
JAS_EXPORT char * jas_strtok (  
    char * str,  
    const char * delim,  
    char ** saveptr)
```

Extract tokens from a string.

This function has a similar behavior as `strtok_r` in the POSIX standard.

Chapter 15

Class Documentation

15.1 `jas_allocator_s` Struct Reference

A memory allocator.

```
#include <jas_malloc.h>
```

Public Member Functions

- void `reserved` (void)

Public Attributes

- void(* `cleanup`)(struct `jas_allocator_s` *allocator)
- void (* `alloc`)(struct `jas_allocator_s` *allocator, size_t size)
- void(* `free`)(struct `jas_allocator_s` *allocator, void *pointer)
- void (* `realloc`)(struct `jas_allocator_s` *allocator, void *pointer, size_t new_size)

15.1.1 Detailed Description

A memory allocator.

15.1.2 Member Function Documentation

15.1.2.1 `reserved()`

```
void jas_allocator_s::reserved (  
    void )
```

For future use.

15.1.3 Member Data Documentation

15.1.3.1 alloc

```
void>(* jas_allocator_s::alloc) (struct jas_allocator_s *allocator, size_t size)
```

Function to allocate memory. This function should have behavior similar to malloc.

15.1.3.2 cleanup

```
void(* jas_allocator_s::cleanup) (struct jas_allocator_s *allocator)
```

Function to clean up the allocator when no longer needed. The allocator cannot be used after the clean-up operation is performed. This function pointer may be null, in which case the clean-up operation is treated as a no-op.

15.1.3.3 free

```
void(* jas_allocator_s::free) (struct jas_allocator_s *allocator, void *pointer)
```

Function to deallocate memory. This function should have behavior similar to free.

15.1.3.4 realloc

```
void>(* jas_allocator_s::realloc) (struct jas_allocator_s *allocator, void *pointer, size_t new←  
_size)
```

Function to reallocate memory. This function should have behavior similar to realloc.

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

- [jas_malloc.h](#)

15.2 jas_cmclrspcconv_t Struct Reference

Color space conversion.

```
#include <jas_cm.h>
```

15.2.1 Detailed Description

Color space conversion.

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

- [jas_cm.h](#)

15.3 `jas_cmcmptfmt_t` Struct Reference

Component format.

```
#include <jas_cm.h>
```

15.3.1 Detailed Description

Component format.

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

- [jas_cm.h](#)

15.4 `jas_cmpixmap_t` Struct Reference

Pixmap (i.e., multicomponent) format.

```
#include <jas_cm.h>
```

15.4.1 Detailed Description

Pixmap (i.e., multicomponent) format.

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

- [jas_cm.h](#)

15.5 `jas_cmprof_t` Struct Reference

```
#include <jas_cm.h>
```

15.5.1 Detailed Description

Color-management (CM) profile.

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

- [jas_cm.h](#)

15.6 `jas_cmpxform_s` Struct Reference

Transform class.

```
#include <jas_cm.h>
```

15.6.1 Detailed Description

Transform class.

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

- [jas_cm.h](#)

15.7 `jas_cmpxformops_t` Struct Reference

Transform operations.

```
#include <jas_cm.h>
```

15.7.1 Detailed Description

Transform operations.

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

- [jas_cm.h](#)

15.8 `jas_cmpxformseq_t` Struct Reference

Primitive transform sequence class.

```
#include <jas_cm.h>
```

15.8.1 Detailed Description

Primitive transform sequence class.

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

- [jas_cm.h](#)

15.9 jas_cmshaplut_t Struct Reference

Shaper look-up table (LUT).

```
#include <jas_cm.h>
```

15.9.1 Detailed Description

Shaper look-up table (LUT).

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

- [jas_cm.h](#)

15.10 jas_cmshapmat_t Struct Reference

Shaper matrix.

```
#include <jas_cm.h>
```

15.10.1 Detailed Description

Shaper matrix.

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

- [jas_cm.h](#)

15.11 jas_cmshapmatlut_t Struct Reference

Shaper matrix look-up table (LUT).

```
#include <jas_cm.h>
```

15.11.1 Detailed Description

Shaper matrix look-up table (LUT).

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

- [jas_cm.h](#)

15.12 `jas_cmxform_t` Struct Reference

Primitive transform class.

```
#include <jas_cm.h>
```

15.12.1 Detailed Description

Primitive transform class.

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

- [jas_cm.h](#)

15.13 `jas_image_cmpt_t` Struct Reference

Image component class.

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

15.13.1 Detailed Description

Image component class.

Warning

Library users should never directly access any of the members of this class. The functions/macros provided by the Jasper library API should always be used.

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

- [jas_image.h](#)

15.14 `jas_image_cmptparm_t` Struct Reference

Component parameters class.

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

15.14.1 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](#)

15.15 `jas_image_fmt_t` Struct Reference

Entry in image format table.

Public Attributes

- `const char *` [name](#)
- `const char *` [desc](#)
- `const char *` [exts](#)
- `const` [jas_image_fmtops_t](#) `ops`
- `int` [enabled](#)

15.15.1 Detailed Description

Entry in image format table.

15.15.2 Member Data Documentation

15.15.2.1 `desc`

```
const char* jas_image_fmt_t::desc
```

A short description of the format.

15.15.2.2 `enabled`

```
int jas_image_fmt_t::enabled
```

A boolean flag indicating if the format is enabled.

15.15.2.3 exts

```
const char* jas_image_fmt_t::exts
```

A whitespace delimited list of file extensions associated with the format.

15.15.2.4 name

```
const char* jas_image_fmt_t::name
```

A unique name identifying the format.

15.15.2.5 ops

```
const jas_image_fmtops_t jas_image_fmt_t::ops
```

The operations for the format (e.g., encode, decode, and validate).

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

- jas_init.c

15.16 jas_image_fmtinfo_t Struct Reference

Image format information.

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

Public Attributes

- int [id](#)
- char * [name](#)
- char ** [exts](#)
- int [enabled](#)
- char * [desc](#)
- [jas_image_fmtops_t](#) [ops](#)

15.16.1 Detailed Description

Image format information.

15.16.2 Member Data Documentation

15.16.2.1 desc

```
char* jas_image_fmtinfo_t::desc
```

A brief description of the format.

15.16.2.2 enabled

```
int jas_image_fmtinfo_t::enabled
```

A boolean flag indicating if this format is enabled.

15.16.2.3 exts

```
char** jas_image_fmtinfo_t::exts
```

The table of file name extensions associated with this format.

15.16.2.4 id

```
int jas_image_fmtinfo_t::id
```

The ID for this format.

15.16.2.5 name

```
char* jas_image_fmtinfo_t::name
```

The name by which this format is identified.

15.16.2.6 ops

```
jas_image_fmtops_t jas_image_fmtinfo_t::ops
```

The operations for this format.

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

- [jas_image.h](#)

15.17 jas_image_fmtops_t Struct Reference

Image format-dependent operations.

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

Public Attributes

- [jas_image_t](#) *(* [decode](#))(jas_stream_t *in, const char *opts)
- int(* [encode](#))(jas_image_t *image, jas_stream_t *out, const char *opts)
- int(* [validate](#))(jas_stream_t *in)

15.17.1 Detailed Description

Image format-dependent operations.

15.17.2 Member Data Documentation

15.17.2.1 decode

```
jas_image_t *(* jas_image_fmtops_t::decode) (jas_stream_t *in, const char *opts)
```

Decode image data from a stream.

15.17.2.2 encode

```
int(* jas_image_fmtops_t::encode) (jas_image_t *image, jas_stream_t *out, const char *opts)
```

Encode image data to a stream.

15.17.2.3 validate

```
int(* jas_image_fmtops_t::validate) (jas_stream_t *in)
```

Determine if stream data is in a particular format.

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

- [jas_image.h](#)

15.18 `jas_image_t` Struct Reference

Image class.

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

15.18.1 Detailed Description

Image class.

Warning

Library users should never directly access any of the members of this class. The functions/macros provided by the JasPer library API should always be used.

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

- [jas_image.h](#)

15.19 `jas_logtype_t` Struct Reference

Type used for the log type.

```
#include <jas_log.h>
```

15.19.1 Detailed Description

Type used for the log type.

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

- [jas_log.h](#)

15.20 `jas_matrix_t` Struct Reference

Matrix type.

```
#include <jas_seq.h>
```

15.20.1 Detailed Description

Matrix type.

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

- [jas_seq.h](#)

15.21 jas_opt_t Struct Reference

Command line option type.

```
#include <jas_getopt.h>
```

15.21.1 Detailed Description

Command line option type.

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

- [jas_getopt.h](#)

15.22 jas_seq2d_t Struct Reference

Two-dimensional sequence type.

```
#include <jas_seq.h>
```

15.22.1 Detailed Description

Two-dimensional sequence type.

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

- [jas_seq.h](#)

15.23 jas_seq_t Struct Reference

One-dimensional sequence type.

```
#include <jas_seq.h>
```

15.23.1 Detailed Description

One-dimensional sequence type.

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

- [jas_seq.h](#)

15.24 `jas_std_allocator_t` Struct Reference

The standard library allocator (i.e., a wrapper for malloc and friends).

```
#include <jas_malloc.h>
```

15.24.1 Detailed Description

The standard library allocator (i.e., a wrapper for malloc and friends).

Essentially, `jas_std_allocator_t` can be thought of as having an inheritance relationship with `jas_allocator_t`. In particular, `jas_std_allocator_t` is derived from `jas_allocator_t`.

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

- [jas_malloc.h](#)

15.25 `jas_stream_t` Struct Reference

I/O stream object.

```
#include <jas_stream.h>
```

15.25.1 Detailed Description

I/O stream object.

Warning

Library users should never directly access any of the members of this class. The functions/macros provided by the Jasper library API should always be used.

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

- [jas_stream.h](#)

15.26 jas_taginfo_t Struct Reference

Tag information type.

```
#include <jas_tvp.h>
```

15.26.1 Detailed Description

Tag information type.

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

- [jas_tvp.h](#)

15.27 jas_tmr_t Struct Reference

Timer type.

```
#include <jas_tmr.h>
```

15.27.1 Detailed Description

Timer type.

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

- [jas_tmr.h](#)

15.28 jas_tvparser_t Struct Reference

Tag-value parser type.

```
#include <jas_tvp.h>
```

15.28.1 Detailed Description

Tag-value parser type.

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

- [jas_tvp.h](#)

Chapter 16

File Documentation

16.1 bmp_cod.h

```
00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
```

```

00052 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060 *
00061 * __END_OF_JASPER_LICENSE__
00062 */
00063
00064 /*
00065 * Windows Bitmap File Library
00066 *
00067 * $Id$
00068 */
00069
00070 #ifndef BMP_COD_H
00071 #define BMP_COD_H
00072
00073 /*****
00074 * Includes.
00075 *****/
00076
00077 #include "jasper/jas_types.h"
00078
00079 /*****
00080 * Constants and macros.
00081 *****/
00082
00083 #define BMP_MAGIC 0x4d42
00084 /* The signature for a BMP file. */
00085
00086 #define BMP_HDRLEN 14
00087 /* The nominal header length. */
00088
00089 #define BMP_INFOLEN 40
00090 /* The nominal info length. */
00091
00092 #define BMP_PALLEN(info) ((info)->numcolors * 4)
00093 /* The length of the palette. */
00094
00095 #define BMP_HASPAL(info) ((info)->numcolors > 0)
00096 /* Is this a palettized image? */
00097
00098 /* Encoding types. */
00099 #define BMP_ENC_RGB 0 /* No special encoding. */
00100 #define BMP_ENC_RLE8 1 /* Run length encoding. */
00101 #define BMP_ENC_RLE4 2 /* Run length encoding. */
00102
00103 /*****
00104 * Types.
00105 *****/
00106
00107 /* BMP header. */
00108 typedef struct {
00109
00110     int_fast16_t magic;
00111     /* The signature (a.k.a. the magic number). */
00112
00113     int_fast32_t siz;
00114     /* The size of the file in 32-bit words. */
00115
00116     int_fast16_t reserved1;
00117     /* Ask Bill Gates what this is all about. */
00118
00119     int_fast16_t reserved2;
00120     /* Ditto. */
00121
00122     int_fast32_t off;
00123     /* The offset of the bitmap data from the bitmap file header in bytes. */
00124
00125 } bmp_hdr_t;
00126
00127 /* Palette entry. */
00128 typedef struct {
00129
00130     int_fast16_t red;
00131     /* The red component. */
00132

```



```

00133         int_fast16_t grn;
00134         /* The green component. */
00135
00136         int_fast16_t blu;
00137         /* The blue component. */
00138
00139         int_fast16_t res;
00140         /* Reserved. */
00141
00142     } bmp_palent_t;
00143
00144     /* BMP info. */
00145     typedef struct {
00146
00147         int_fast32_t len;
00148         /* The length of the bitmap information header in bytes. */
00149
00150         int_fast32_t width;
00151         /* The width of the bitmap in pixels. */
00152
00153         int_fast32_t height;
00154         /* The height of the bitmap in pixels. */
00155
00156         int_fast8_t topdown;
00157         /* The bitmap data is specified in top-down order. */
00158
00159         int_fast16_t numplanes;
00160         /* The number of planes. This must be set to a value of one. */
00161
00162         int_fast16_t depth;
00163         /* The number of bits per pixel. */
00164
00165         int_fast32_t enctype;
00166         /* The type of compression used. */
00167
00168         int_fast32_t siz;
00169         /* The size of the image in bytes. */
00170
00171         int_fast32_t hres;
00172         /* The horizontal resolution in pixels/metre. */
00173
00174         int_fast32_t vres;
00175         /* The vertical resolution in pixels/metre. */
00176
00177         int_fast32_t numcolors;
00178         /* The number of color indices used by the bitmap. */
00179
00180         int_fast32_t mincolors;
00181         /* The number of color indices important for displaying the bitmap. */
00182
00183         bmp_palent_t *palents;
00184         /* The colors should be listed in order of importance. */
00185
00186     } bmp_info_t;
00187
00188     /*****
00189     * Functions and macros.
00190     *****/
00191
00192     #define bmp_issupported(hdr, info) \
00193         ((hdr)->magic == BMP_MAGIC && !(hdr)->reserved1 && \
00194          !(hdr)->reserved2 && (info)->numplanes == 1 && \
00195          ((info)->depth == 8 || (info)->depth == 24) && \
00196          (info)->enctype == BMP_ENC_RGB)
00197     /* Is this type of BMP file supported? */
00198
00199     #define bmp_haspal(info) \
00200         ((info)->depth == 8)
00201     /* Is there a palette? */
00202
00203     int bmp_numcmpts(bmp_info_t *info);
00204     /* Get the number of components. */
00205
00206     bmp_info_t *bmp_info_create(void);
00207     /* Create BMP information. */
00208
00209     void bmp_info_destroy(bmp_info_t *info);
00210     /* Destroy BMP information. */
00211
00212     int bmp_isgrayscalepal(bmp_palent_t *palents, int numpalents);
00213     /* Does the specified palette correspond to a grayscale image? */

```

```
00214
00215 #endif
```

16.2 bmp_enc.h

```
00001 /*
00002  * Copyright (c) 2002 Michael David Adams.
00003  * All rights reserved.
00004  */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007  *
00008  * JasPer License Version 2.0
00009  *
00010  * Copyright (c) 2001-2006 Michael David Adams
00011  * Copyright (c) 1999-2000 Image Power, Inc.
00012  * Copyright (c) 1999-2000 The University of British Columbia
00013  *
00014  * All rights reserved.
00015  *
00016  * Permission is hereby granted, free of charge, to any person (the
00017  * "User") obtaining a copy of this software and associated documentation
00018  * files (the "Software"), to deal in the Software without restriction,
00019  * including without limitation the rights to use, copy, modify, merge,
00020  * publish, distribute, and/or sell copies of the Software, and to permit
00021  * persons to whom the Software is furnished to do so, subject to the
00022  * following conditions:
00023  *
00024  * 1. The above copyright notices and this permission notice (which
00025  * includes the disclaimer below) shall be included in all copies or
00026  * substantial portions of the Software.
00027  *
00028  * 2. The name of a copyright holder shall not be used to endorse or
00029  * promote products derived from the Software without specific prior
00030  * written permission.
00031  *
00032  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037  * PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00054  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058  *
00059  * __END_OF_JASPER_LICENSE__
00060  */
00061
00062 #ifndef BMP_ENC_H
00063 #define BMP_ENC_H
00064
00065 typedef struct {
00066
00067     int numcmpts;
00068     int cmpts[4];
00069
00070 } bmp_enc_t;
00071
00072 #endif
```

16.3 jas_cm.h File Reference

JasPer Color Management.

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

Classes

- struct [jas_cmcmptfmt_t](#)
Component format.
- struct [jas_cmpixmap_t](#)
Pixmap (i.e., multicomponent) format.
- struct [jas_cmpxformops_t](#)
Transform operations.
- struct [jas_cmshapmatlut_t](#)
Shaper matrix look-up table (LUT).
- struct [jas_cmshapmat_t](#)
Shaper matrix.
- struct [jas_cmshaplut_t](#)
Shaper look-up table (LUT).
- struct [jas_cmclrspcconv_t](#)
Color space conversion.
- struct [jas_cmpxform_s](#)
Transform class.
- struct [jas_cmpxformseq_t](#)
Primitive transform sequence class.
- struct [jas_cmxfom_t](#)
Primitive transform class.
- struct [jas_cmprof_t](#)

Macros

- #define [JAS_CMXFORM_NUMINTENTS](#) 4
Number of rendering intents.
- #define [jas_clrspc_create](#)(fam, mbr)
Create a color space.
- #define [jas_clrspc_fam](#)(clrspc)
Get the family of a color space.
- #define [jas_clrspc_mbr](#)(clrspc)
Get the (family) member of a color space.
- #define [jas_clrspc_isgeneric](#)(clrspc)
Test if a color space is generic.
- #define [jas_clrspc_isunknown](#)(clrspc)

- *Test if a color space is unknown.*
- #define `JAS_CLRSPC_FAM_UNKNOWN` 0
- *Color space families.*
- #define `JAS_CLRSPC_UNKNOWN` `JAS_CLRSPC_UNKNOWNMASK`
- *Specific color spaces.*
- #define `JAS_CLRSPC_GENRGB` `jas_clrspc_create`(`JAS_CLRSPC_FAM_RGB`, 0)
- *Generic color spaces.*
- #define `jas_cmprof_clrspc`(`prof`)
- *Get the color space associated with a color-management profile.*

Typedefs

- typedef unsigned `jas_clrspc_t`
- *Color space.*
- typedef double `jas_cmreal_t`
- typedef struct `jas_cmpxform_s` `jas_cmpxform_t`
- *Transform class.*

Enumerations

- enum `jas_cmxform_op_t`
- *Transform operations.*
- enum `jas_cmxform_intent_t`
- *Rendering intents.*
- enum `jas_cmxform_optm_t`
- *Transform optimization.*

Functions

- JAS_EXPORT `jas_cmprof_t` * `jas_cmprof_createfromiccp`(const `jas_iccp` *`iccp`)
- *Create a color-management profile from an ICC profile.*
- JAS_EXPORT `jas_cmprof_t` * `jas_cmprof_createfromclrspc`(`jas_clrspc_t` `clrspc`)
- *Create a color-management profile from a color space.*
- JAS_EXPORT void `jas_cmprof_destroy`(`jas_cmprof_t` *`prof`)
- *Destroy a color-management profile.*
- JAS_EXPORT `jas_cmprof_t` * `jas_cmprof_copy`(const `jas_cmprof_t` *`prof`)
- *Copy a color-management profile.*
- JAS_EXPORT `jas_iccp` * `jas_iccp_createfromcmprof`(const `jas_cmprof_t` *`prof`)
- *Create a ICC profile from a CM profile.*
- JAS_EXPORT `jas_cmxform_t` * `jas_cmxform_create`(const `jas_cmprof_t` *`inprof`, const `jas_cmprof_t` *`outprof`, const `jas_cmprof_t` *`proofprof`, `jas_cmxform_op_t` `op`, `jas_cmxform_intent_t` `intent`, `jas_cmxform_optm_t` `optimize`)
- *Create a transform from a CM profile.*
- JAS_EXPORT void `jas_cmxform_destroy`(`jas_cmxform_t` *`xform`)
- *Destroy a transform.*
- JAS_EXPORT int `jas_cmxform_apply`(const `jas_cmxform_t` *`xform`, const `jas_cmpixmap_t` *`in`, `jas_cmpixmap_t` *`out`)
- *Apply a transform to data.*
- unsigned `jas_clrspc_numchans`(`jas_clrspc_t` `clrspc`)
- *Get the number of channels associated with a particular color space.*

16.3.1 Detailed Description

JasPer Color Management.

16.4 jas_cm.h

[Go to the documentation of this file.](#)

```

00001 /*
00002  * Copyright (c) 2002-2003 Michael David Adams.
00003  * All rights reserved.
00004  */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007  *
00008  * JasPer License Version 2.0
00009  *
00010  * Copyright (c) 2001-2006 Michael David Adams
00011  * Copyright (c) 1999-2000 Image Power, Inc.
00012  * Copyright (c) 1999-2000 The University of British Columbia
00013  *
00014  * All rights reserved.
00015  *
00016  * Permission is hereby granted, free of charge, to any person (the
00017  * "User") obtaining a copy of this software and associated documentation
00018  * files (the "Software"), to deal in the Software without restriction,
00019  * including without limitation the rights to use, copy, modify, merge,
00020  * publish, distribute, and/or sell copies of the Software, and to permit
00021  * persons to whom the Software is furnished to do so, subject to the
00022  * following conditions:
00023  *
00024  * 1. The above copyright notices and this permission notice (which
00025  * includes the disclaimer below) shall be included in all copies or
00026  * substantial portions of the Software.
00027  *
00028  * 2. The name of a copyright holder shall not be used to endorse or
00029  * promote products derived from the Software without specific prior
00030  * written permission.
00031  *
00032  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00054  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058  *
00059  * __END_OF_JASPER_LICENSE__
00060  */
00061
00062 #ifndef JAS_CM_H
00063 #define JAS_CM_H
00064
00065 /*****\

```

```

00071 *
00072 \*****
00073
00074 /* The configuration header file should be included first. */
00075 #include <jasper/jas_config.h>
00076
00077 #include <jasper/jas_types.h>
00078 #include <jasper/jas_icc.h>
00079
00080 #ifdef __cplusplus
00081 extern "C" {
00082 #endif
00083
00084
00085
00086
00087
00088
00089 /*****
00090 * Types and Macros.
00091 \*****
00092
00093 typedef unsigned jas_clrspc_t;
00094
00095 typedef enum {
00096     JAS_CMXFORM_OP_FWD = 0,
00097     JAS_CMXFORM_OP_REV = 1,
00098     JAS_CMXFORM_OP_PROOF = 2,
00099     JAS_CMXFORM_OP_GAMUT = 3,
00100 } jas_cmxform_op_t;
00101
00102 typedef enum {
00103     JAS_CMXFORM_INTENT_PER = 0,
00104     JAS_CMXFORM_INTENT_RELCLR = 1,
00105     JAS_CMXFORM_INTENT_ABSCLR = 2,
00106     JAS_CMXFORM_INTENT_SAT = 3,
00107 } jas_cmxform_intent_t;
00108
00109 #define JAS_CMXFORM_NUMINTENTS 4
00110
00111 typedef enum {
00112     JAS_CMXFORM_OPTM_SPEED = 0,
00113     JAS_CMXFORM_OPTM_SIZE = 1,
00114     JAS_CMXFORM_OPTM_ACC = 2,
00115 } jas_cmxform_optm_t;
00116
00117 #define jas_clrspc_create(fam, mbr) (((fam) << 8) | (mbr))
00118
00119 #define jas_clrspc_fam(clrspc) ((clrspc) >> 8)
00120
00121 #define jas_clrspc_mbr(clrspc) ((clrspc) & 0xff)
00122
00123 #define jas_clrspc_isgeneric(clrspc) (!jas_clrspc_mbr(clrspc))
00124
00125 #define jas_clrspc_isunknown(clrspc) ((clrspc) & JAS_CLRSPC_UNKNOWNMASK)
00126
00127 #define JAS_CLRSPC_UNKNOWNMASK 0x4000
00128
00129 #define JAS_CLRSPC_FAM_UNKNOWN 0
00130 #define JAS_CLRSPC_FAM_XYZ 1
00131 #define JAS_CLRSPC_FAM_LAB 2
00132 #define JAS_CLRSPC_FAM_GRAY 3
00133 #define JAS_CLRSPC_FAM_RGB 4
00134 #define JAS_CLRSPC_FAM_YCBCR 5
00135
00136 #define JAS_CLRSPC_UNKNOWN JAS_CLRSPC_UNKNOWNMASK
00137 #define JAS_CLRSPC_CIEXYZ jas_clrspc_create(JAS_CLRSPC_FAM_XYZ, 1)
00138 #define JAS_CLRSPC_CIELAB jas_clrspc_create(JAS_CLRSPC_FAM_LAB, 1)
00139 #define JAS_CLRSPC_SGRAY jas_clrspc_create(JAS_CLRSPC_FAM_GRAY, 1)
00140 #define JAS_CLRSPC_SRGB jas_clrspc_create(JAS_CLRSPC_FAM_RGB, 1)
00141 #define JAS_CLRSPC_SYCBCR jas_clrspc_create(JAS_CLRSPC_FAM_YCBCR, 1)
00142
00143 #define JAS_CLRSPC_GENRGB jas_clrspc_create(JAS_CLRSPC_FAM_RGB, 0)
00144 #define JAS_CLRSPC_GENGRAY jas_clrspc_create(JAS_CLRSPC_FAM_GRAY, 0)
00145 #define JAS_CLRSPC_GENYCBCR jas_clrspc_create(JAS_CLRSPC_FAM_YCBCR, 0)
00146
00147 #define JAS_CLRSPC_CHANIND_YCBCR_Y 0
00148 #define JAS_CLRSPC_CHANIND_YCBCR_CB 1
00149 #define JAS_CLRSPC_CHANIND_YCBCR_CR 2
00150
00151 #define JAS_CLRSPC_CHANIND_RGB_R 0
00152 #define JAS_CLRSPC_CHANIND_RGB_G 1
00153 #define JAS_CLRSPC_CHANIND_RGB_B 2
00154

```

```

00205 #define JAS_CLRSPC_CHANIND_GRAY_Y      0
00206
00210 typedef double jas_cmreal_t;
00211
00212 struct jas_cmpxform_s;
00213
00218 typedef struct {
00219     long *buf;
00220     unsigned prec;
00221     int sgnd;
00222     unsigned width;
00223     unsigned height;
00224 } jas_cmcmptfmt_t;
00225
00230 typedef struct {
00231     unsigned numcmpts;
00232     jas_cmcmptfmt_t *cmptfmts;
00233 } jas_cmpixmap_t;
00234
00239 typedef struct {
00240     void (*destroy)(struct jas_cmpxform_s *pxform);
00241     int (*apply)(const struct jas_cmpxform_s *pxform, const jas_cmreal_t *in, jas_cmreal_t *out,
00242                 unsigned cnt);
00242     void (*dump)(struct jas_cmpxform_s *pxform);
00243 } jas_cmpxformops_t;
00244
00249 typedef struct {
00250     jas_cmreal_t *data;
00251     unsigned size;
00252 } jas_cmshapmatlut_t;
00253
00258 typedef struct {
00259     int mono;
00260     int order;
00261     int useluts;
00262     int usemat;
00263     jas_cmshapmatlut_t luts[3];
00264     jas_cmreal_t mat[3][4];
00265 } jas_cmshapmat_t;
00266
00271 typedef struct {
00272     int order;
00273 } jas_cmshaplut_t;
00274
00279 typedef struct {
00280     unsigned inclrspc;
00281     unsigned outclrspc;
00282 } jas_cmclrspcconv_t;
00283
00288 typedef struct jas_cmpxform_s {
00289     unsigned refcnt;
00290     const jas_cmpxformops_t *ops;
00291     unsigned numinchans;
00292     unsigned numoutchans;
00293     union {
00294         max_align_t dummy;
00295         jas_cmshapmat_t shapmat;
00296         jas_cmshaplut_t shaplut;
00297         jas_cmclrspcconv_t clrspcconv;
00298     } data;
00299 } jas_cmpxform_t;
00300
00305 typedef struct {
00306     unsigned numpxforms;
00307     unsigned maxpxforms;
00308     jas_cmpxform_t **pxforms;
00309 } jas_cmpxformseq_t;
00310
00315 typedef struct {
00316     unsigned numinchans;
00317     unsigned numoutchans;
00318     jas_cmpxformseq_t *pxformseq;
00319 } jas_cmxfm_t;
00320
00321 #define JAS_CMPROF_TYPE_DEV      1
00322 #define JAS_CMPROF_TYPE_CLRSPC  2
00323
00324 #define JAS_CMPROF_NUMPXFORMSEQS 13
00325
00329 typedef struct {
00330     jas_clrspc_t clrspc;

```

```

00331         unsigned numchans;
00332         unsigned refclrspc;
00333         unsigned numrefchans;
00334         jas_iccprof_t *iccprof;
00335         jas_cmxformseq_t *pxformseqs[JAS_CMPROF_NUMPXFORMSEQS];
00336     } jas_cmprof_t;
00337
00338     /*****
00339     *
00340     \*****/
00341
00342     #if 0
00343     typedef int_fast32_t jas_cmattrname_t;
00344     typedef int_fast32_t jas_cmattrval_t;
00345     typedef int_fast32_t jas_cmattrtype_t;
00346     /* Load a profile. */
00347     int jas_cmprof_load(jas_cmprof_t *prof, jas_stream_t *in, unsigned fmt);
00348     /* Save a profile. */
00349     int jas_cmprof_save(jas_cmprof_t *prof, jas_stream_t *out, unsigned fmt);
00350     /* Set an attribute of a profile. */
00351     int jas_cm_prof_setattr(jas_cm_prof_t *prof, jas_cm_attrname_t name, void *val);
00352     /* Get an attribute of a profile. */
00353     void *jas_cm_prof_getattr(jas_cm_prof_t *prof, jas_cm_attrname_t name);
00354     #endif
00355
00356     /*****
00357     * Color-management (CM) profile class.
00358     \*****/
00359
00371     JAS_EXPORT
00372     jas_cmprof_t *jas_cmprof_createfromiccprof(const jas_iccprof_t *iccprof);
00373
00385     JAS_EXPORT
00386     jas_cmprof_t *jas_cmprof_createfromclrspc(jas_clrspc_t clrspc);
00387
00393     JAS_EXPORT
00394     void jas_cmprof_destroy(jas_cmprof_t *prof);
00395
00407     JAS_EXPORT
00408     jas_cmprof_t *jas_cmprof_copy(const jas_cmprof_t *prof);
00409
00421     JAS_EXPORT
00422     jas_iccprof_t *jas_iccprof_createfromcmprof(const jas_cmprof_t *prof);
00423
00424     /*****
00425     * Color-Management (CM) Transform.
00426     \*****/
00427
00438     JAS_EXPORT
00439     jas_cmxform_t *jas_cmxform_create(const jas_cmprof_t *inprof,
00440         const jas_cmprof_t *outprof, const jas_cmprof_t *proofprof,
00441         jas_cmxform_op_t op, jas_cmxform_intent_t intent,
00442         jas_cmxform_optm_t optimize);
00443
00450     JAS_EXPORT
00451     void jas_cmxform_destroy(jas_cmxform_t *xform);
00452
00463     JAS_EXPORT
00464     int jas_cmxform_apply(const jas_cmxform_t *xform, const jas_cmap_t *in,
00465         jas_cmap_t *out);
00466
00467     /*****
00468     * Miscellany.
00469     \*****/
00470
00479     unsigned jas_clrspc_numchans(jas_clrspc_t clrspc);
00480
00489     #define jas_cmprof_clrspc(prof) ((prof)->clrspc)
00490
00494
00495     #ifdef __cplusplus
00496     }
00497     #endif
00498
00499     #endif

```


16.5 jas_compiler.h File Reference

Compiler-related macros.

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

Macros

- `#define JAS_UNUSED(x)`
Indicate that a variable may be unused (in order to avoid a compiler warning).

16.5.1 Detailed Description

Compiler-related macros.

16.5.2 Macro Definition Documentation

16.5.2.1 JAS_UNUSED

```
#define JAS_UNUSED(  
    x)
```

Value:

```
((void) x)
```

Indicate that a variable may be unused (in order to avoid a compiler warning).

16.6 jas_compiler.h

[Go to the documentation of this file.](#)

```
00001 /* __START_OF_JASPER_LICENSE__  
00002 *  
00003 * JasPer License Version 2.0  
00004 *  
00005 * Copyright (c) 2001-2006 Michael David Adams  
00006 * Copyright (c) 1999-2000 Image Power, Inc.  
00007 * Copyright (c) 1999-2000 The University of British Columbia  
00008 *  
00009 * All rights reserved.  
00010 *  
00011 * Permission is hereby granted, free of charge, to any person (the  
00012 * "User") obtaining a copy of this software and associated documentation  
00013 * files (the "Software"), to deal in the Software without restriction,  
00014 * including without limitation the rights to use, copy, modify, merge,  
00015 * publish, distribute, and/or sell copies of the Software, and to permit  
00016 * persons to whom the Software is furnished to do so, subject to the  
00017 * following conditions:  
00018 *  
00019 * 1. The above copyright notices and this permission notice (which  
00020 * includes the disclaimer below) shall be included in all copies or  
00021 * substantial portions of the Software.
```

```

00022 *
00023 * 2. The name of a copyright holder shall not be used to endorse or
00024 * promote products derived from the Software without specific prior
00025 * written permission.
00026 *
00027 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00028 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00029 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00030 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00031 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00032 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00033 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00034 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00035 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00036 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00037 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00038 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00039 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00040 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00041 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00042 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00043 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00044 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00045 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00046 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00047 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00048 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00049 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00050 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00051 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00052 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00053 *
00054 * __END_OF_JASPER_LICENSE__
00055 */
00056
00061
00062 #ifndef JAS_COMPILER_H
00063 #define JAS_COMPILER_H
00064
00065 /* The configuration header file should be included first. */
00066 #include <jasper/jas_config.h>
00067
00068 #ifdef _MSC_VER
00069 #   ifndef __cplusplus
00070 #       undef inline
00071 #       define inline __inline
00072 #   endif
00073 #endif
00074
00075 #if defined(__GNUC__)
00076 #   define JAS_DEPRECATED __attribute__((deprecated))
00077 #else
00078 #   define JAS_DEPRECATED
00079 #endif
00080
00081 #if defined(__GNUC__)
00082 #   define JAS_ATTRIBUTE_CONST __attribute__((const))
00083 #else
00084 #   define JAS_ATTRIBUTE_CONST
00085 #endif
00086
00087 #if defined(__GNUC__)
00088 #   define JAS_ATTRIBUTE_PURE __attribute__((pure))
00089 #else
00090 #   define JAS_ATTRIBUTE_PURE
00091 #endif
00092
00093 #if defined(__GNUC__)
00094 #   define JAS_FORCE_INLINE inline __attribute__((always_inline))
00095 #else
00096 #   define JAS_FORCE_INLINE inline
00097 #endif
00098
00099 #if defined(__GNUC__)
00100 #   if __GNUC__ > 4 || (__GNUC__ == 4 && __GNUC_MINOR__ >= 5)
00101 #       define JAS_UNREACHABLE() __builtin_unreachable()
00102 #   else
00103 #       define JAS_UNREACHABLE()
00104 #   endif
00105 #elif defined(__clang__)
00106 #   define JAS_UNREACHABLE() __builtin_unreachable()

```

```

00107 #elif defined(_MSC_VER)
00108 #     define JAS_UNREACHABLE() __assume(0)
00109 #else
00110 #     define JAS_UNREACHABLE()
00111 #endif
00112
00113 #if defined(__GNUC__)
00114 #     define JAS_LIKELY(x) __builtin_expect (!! (x), 1)
00115 #else
00116 #     define JAS_LIKELY(x) (x)
00117 #endif
00118
00119 #if defined(__GNUC__)
00120 #     define JAS_UNLIKELY(x) __builtin_expect (!! (x), 0)
00121 #else
00122 #     define JAS_UNLIKELY(x) (x)
00123 #endif
00124
00125 #if defined(__GNUC__) && __GNUC__ >= 6
00126 #     define JAS_ATTRIBUTE_DISABLE_UBSAN \
00127         __attribute__((no_sanitize_undefined))
00128 #elif defined(__clang__)
00129 #     define JAS_ATTRIBUTE_DISABLE_UBSAN \
00130         __attribute__((no_sanitize("undefined")))
00131 #else
00132 #     define JAS_ATTRIBUTE_DISABLE_UBSAN
00133 #endif
00134
00135 #ifdef __has_builtin
00136 #define jas_has_builtin(x) __has_builtin(x)
00137 #else
00138 #define jas_has_builtin(x) 0
00139 #endif
00140
00145 #define JAS_UNUSED(x) ((void) x)
00146
00147 #endif

```

16.7 jas_debug.h File Reference

JasPer Debugging-Related Functionality.

```

#include <jasper/jas_config.h>
#include "jasper/jas_init.h"
#include "jasper/jas_debug.h"
#include <stdio.h>
#include <stdarg.h>

```

Macros

- `#define JAS_STRINGIFY(x)`
Convert to a string literal.
- `#define JAS_STRINGIFYX(x)`
Convert to a string literal after macro expansion.

Functions

- JAS_EXPORT void [jas_deprecated](#) (const char *fmt,...)
Warn about the use of deprecated functionality.
- static JAS_DEPRECATED int [jas_getdbglevel](#) (void)
Get the library debug level.
- JAS_EXPORT int [jas_setdbglevel](#) (int dbglevel)
Set the library debug level.
- JAS_EXPORT int [jas_eprintf](#) (const char *fmt,...)
Print formatted text for the standard error stream (i.e., stderr).
- JAS_EXPORT int [jas_logprintf](#) (const char *fmt,...)
Generate a generic log message.
- JAS_EXPORT int [jas_logerrorf](#) (const char *fmt,...)
Generate an error log message.
- JAS_EXPORT int [jas_logwarnf](#) (const char *fmt,...)
Generate a warning log message.
- JAS_EXPORT int [jas_loginf](#) (const char *fmt,...)
Generate an informational log message.
- JAS_EXPORT int [jas_logdebugf](#) (int priority, const char *fmt,...)
Generate a debugging log message.
- int [jas_logmemdump](#) (const void *data, size_t len)
Dump memory.
- JAS_EXPORT int [jas_memdump](#) (FILE *out, const void *data, size_t len)
Dump memory to a stream.

16.7.1 Detailed Description

JasPer Debugging-Related Functionality.

16.7.2 Macro Definition Documentation

16.7.2.1 JAS_STRINGIFY

```
#define JAS_STRINGIFY(  
    x)
```

Value:

```
#x
```

Convert to a string literal.

16.7.2.2 JAS_STRINGIFYX

```
#define JAS_STRINGIFYX(  
    x)
```

Value:

`JAS_STRINGIFY(x)`

Convert to a string literal after macro expansion.

16.7.3 Function Documentation

16.7.3.1 jas_deprecated()

```
JAS_EXPORT void jas_deprecated (  
    const char * fmt,  
    ...)
```

Warn about the use of deprecated functionality.

16.7.3.2 jas_eprintf()

```
JAS_EXPORT int jas_eprintf (  
    const char * fmt,  
    ...)
```

Print formatted text for the standard error stream (i.e., stderr).

16.7.3.3 jas_getdbglevel()

```
JAS_DEPRECATED int jas_getdbglevel (  
    void ) [inline], [static]
```

Get the library debug level.

Deprecated This function is deprecated.

16.7.3.4 jas_logdebugf()

```
JAS_EXPORT int jas_logdebugf (  
    int priority,  
    const char * fmt,  
    ...)
```

Generate a debugging log message.

16.7.3.5 `jas_logerrorf()`

```
JAS_EXPORT int jas_logerrorf (  
    const char * fmt,  
    ...)
```

Generate an error log message.

16.7.3.6 `jas_loginfof()`

```
JAS_EXPORT int jas_loginfof (  
    const char * fmt,  
    ...)
```

Generate an informational log message.

16.7.3.7 `jas_logmemdump()`

```
int jas_logmemdump (  
    const void * data,  
    size_t len)
```

Dump memory.

16.7.3.8 `jas_logprintf()`

```
JAS_EXPORT int jas_logprintf (  
    const char * fmt,  
    ...)
```

Generate a generic log message.

16.7.3.9 `jas_logwarnf()`

```
JAS_EXPORT int jas_logwarnf (  
    const char * fmt,  
    ...)
```

Generate a warning log message.

16.7.3.10 `jas_memdump()`

```
JAS_EXPORT int jas_memdump (  
    FILE * out,  
    const void * data,  
    size_t len)
```

Dump memory to a stream.

16.7.3.11 jas_setdbglevel()

```
JAS_EXPORT int jas_setdbglevel (
    int dbglevel)
```

Set the library debug level.

16.8 jas_debug.h

[Go to the documentation of this file.](#)

```
00001 /*
00002  * Copyright (c) 2001-2002 Michael David Adams.
00003  * All rights reserved.
00004  */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007  *
00008  * JasPer License Version 2.0
00009  *
00010  * Copyright (c) 2001-2006 Michael David Adams
00011  * Copyright (c) 1999-2000 Image Power, Inc.
00012  * Copyright (c) 1999-2000 The University of British Columbia
00013  *
00014  * All rights reserved.
00015  *
00016  * Permission is hereby granted, free of charge, to any person (the
00017  * "User") obtaining a copy of this software and associated documentation
00018  * files (the "Software"), to deal in the Software without restriction,
00019  * including without limitation the rights to use, copy, modify, merge,
00020  * publish, distribute, and/or sell copies of the Software, and to permit
00021  * persons to whom the Software is furnished to do so, subject to the
00022  * following conditions:
00023  *
00024  * 1. The above copyright notices and this permission notice (which
00025  * includes the disclaimer below) shall be included in all copies or
00026  * substantial portions of the Software.
00027  *
00028  * 2. The name of a copyright holder shall not be used to endorse or
00029  * promote products derived from the Software without specific prior
00030  * written permission.
00031  *
00032  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00054  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058  *
00059  * __END_OF_JASPER_LICENSE__
00060  */
00061
```

```

00066
00067 #ifndef JAS_DEBUG_H
00068 #define JAS_DEBUG_H
00069
00070 /*****
00071  * Includes.
00072  *****/
00073
00074 /* The configuration header file should be included first. */
00075 #include <jasper/jas_config.h>
00076
00077 #include "jasper/jas_init.h"
00078 #include "jasper/jas_debug.h"
00079
00080 #include <stdio.h>
00081 #include <stdarg.h>
00082
00083 #ifdef __cplusplus
00084 extern "C" {
00085 #endif
00086
00087 /*****
00088  * Macros and functions.
00089  *****/
00090
00091 /* Output debugging information to standard error provided that the debug
00092    level is set sufficiently high. */
00093 #if !defined(NDEBUG)
00094 #define JAS_DBGLOG(n, x) \
00095     ((jas_get_debug_level() >= (n)) ? (jas_eprintf x) : 0)
00096 #else
00097 #define JAS_DBGLOG(n, x)
00098 #endif
00099
00100 #if !defined(NDEBUG)
00101 #define JAS_LOGDEBUGF(n, ...) \
00102     ((jas_get_debug_level() >= (n)) ? jas_logdebugf((n), __VA_ARGS__) : 0)
00103 #else
00104 #define JAS_LOGDEBUGF(n, ...)
00105 #endif
00106
00111 JAS_EXPORT
00112 void jas_deprecated(const char *fmt, ...);
00113
00121 JAS_DEPRECATED
00122 static inline
00123 int jas_getdbglevel(void)
00124 {
00125     jas_deprecated("jas_getdbglevel is deprecated\n");
00126     return jas_get_debug_level();
00127 }
00128
00133 JAS_EXPORT
00134 int jas_setdbglevel(int dbglevel);
00135
00140 JAS_EXPORT
00141 int jas_eprintf(const char *fmt, ...);
00142
00147 JAS_EXPORT
00148 int jas_logprintf(const char *fmt, ...);
00149
00154 JAS_EXPORT
00155 int jas_logerrorf(const char *fmt, ...);
00156
00161 JAS_EXPORT
00162 int jas_logwarnf(const char *fmt, ...);
00163
00168 JAS_EXPORT
00169 int jas_loginf(const char *fmt, ...);
00170
00175 JAS_EXPORT
00176 int jas_logdebugf(int priority, const char *fmt, ...);
00177
00182 int jas_logmemdump(const void *data, size_t len);
00183
00188 JAS_EXPORT
00189 int jas_memdump(FILE *out, const void *data, size_t len);
00190
00195 #define JAS_STRINGIFY(x) #x
00196
00201 #define JAS_STRINGIFYX(x) JAS_STRINGIFY(x)

```



```
00202
00203 #ifdef __cplusplus
00204 }
00205 #endif
00206
00207 #endif
```

16.9 jas_dll.h File Reference

Shared Library Macros.

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

16.9.1 Detailed Description

Shared Library Macros.

16.10 jas_dll.h

[Go to the documentation of this file.](#)

```
00001 /* __START_OF_JASPER_LICENSE__
00002  *
00003  * JasPer License Version 2.0
00004  *
00005  * Copyright (c) 2001-2006 Michael David Adams
00006  * Copyright (c) 1999-2000 Image Power, Inc.
00007  * Copyright (c) 1999-2000 The University of British Columbia
00008  *
00009  * All rights reserved.
00010  *
00011  * Permission is hereby granted, free of charge, to any person (the
00012  * "User") obtaining a copy of this software and associated documentation
00013  * files (the "Software"), to deal in the Software without restriction,
00014  * including without limitation the rights to use, copy, modify, merge,
00015  * publish, distribute, and/or sell copies of the Software, and to permit
00016  * persons to whom the Software is furnished to do so, subject to the
00017  * following conditions:
00018  *
00019  * 1. The above copyright notices and this permission notice (which
00020  * includes the disclaimer below) shall be included in all copies or
00021  * substantial portions of the Software.
00022  *
00023  * 2. The name of a copyright holder shall not be used to endorse or
00024  * promote products derived from the Software without specific prior
00025  * written permission.
00026  *
00027  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00028  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00029  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00030  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00031  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00032  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00033  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00034  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00035  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00036  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00037  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00038  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00039  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00040  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
```

```

00041  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00042  * PROPERTY RIGHTS OR OTHERWISE.  AS A CONDITION TO EXERCISING THE RIGHTS
00043  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00044  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY.  THE SOFTWARE
00045  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00046  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00047  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00048  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00049  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00050  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00051  * RISK ACTIVITIES").  THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00052  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00053  *
00054  * __END_OF_JASPER_LICENSE__
00055  */
00056
00061
00062 #ifndef JAS_DLL_H
00063 #define JAS_DLL_H
00064
00065 /* The configuration header file should be included first. */
00066 #include <jasper/jas_config.h>
00067
00068 #include <jasper/jas_export_cmake.h>
00069
00070 /* For backward compatibility only. */
00071 #define JAS_DLLEXPORT JAS_EXPORT
00072 /* For backward compatibility only. */
00073 #define JAS_DLLLOCAL JAS_LOCAL
00074
00075 #endif

```

16.11 jas_fix.h File Reference

JasPer Fixed-Point Number Class.

```

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

```

Macros

- `#define JAS_FIX_ZERO(fix_t, fracbits)`
- `#define JAS_FIX_ONE(fix_t, fracbits)`
- `#define JAS_FIX_HALF(fix_t, fracbits)`
- `#define JAS_INTTOFIX(fix_t, fracbits, x)`
- `#define JAS_FIXTOINT(fix_t, fracbits, x)`
- `#define JAS_FIXTODBL(fix_t, fracbits, x)`
- `#define JAS_DBLTOFIX(fix_t, fracbits, x)`
- `#define JAS_FIX_ADD JAS_FIX_ADD_FAST`
- `#define JAS_FIX_ADD_FAST(fix_t, fracbits, x, y)`
- `#define JAS_FIX_ADD_OFLOW(fix_t, fracbits, x, y)`
- `#define JAS_FIX_MUL JAS_FIX_MUL_FAST`
- `#define JAS_FIX_MUL_FAST(fix_t, fracbits, bigfix_t, x, y)`
- `#define JAS_FIX_MUL_OFLOW(fix_t, fracbits, bigfix_t, x, y)`
- `#define JAS_FIX_MULBYINT JAS_FIX_MULBYINT_FAST`
- `#define JAS_FIX_MULBYINT_FAST(fix_t, fracbits, x, y)`
- `#define JAS_FIX_MULBYINT_OFLOW(fix_t, fracbits, x, y)`
- `#define JAS_FIX_DIV JAS_FIX_DIV_FAST`

- `#define JAS_FIX_DIV_FAST(fix_t, fracbits, bigfix_t, x, y)`
- `#define JAS_FIX_DIV_UFLOW(fix_t, fracbits, bigfix_t, x, y)`
- `#define JAS_FIX_NEG JAS_FIX_NEG_FAST`
- `#define JAS_FIX_NEG_FAST(fix_t, fracbits, x)`
- `#define JAS_FIX_NEG_OFLOW(fix_t, fracbits, x)`
- `#define JAS_FIX_ASJ JAS_FIX_ASJ_FAST`
- `#define JAS_FIX_ASJ_FAST(fix_t, fracbits, x, n)`
- `#define JAS_FIX_ASJ_OFLOW(fix_t, fracbits, x, n)`
- `#define JAS_FIX_ASR JAS_FIX_ASR_FAST`
- `#define JAS_FIX_ASR_FAST(fix_t, fracbits, x, n)`
- `#define JAS_FIX_ASR_UFLOW(fix_t, fracbits, x, n)`
- `#define JAS_FIX_SUB(fix_t, fracbits, x, y)`
- `#define JAS_FIX_PLUSEQ(fix_t, fracbits, x, y)`
- `#define JAS_FIX_MINUSEQ(fix_t, fracbits, x, y)`
- `#define JAS_FIX_MULEQ(fix_t, fracbits, bigfix_t, x, y)`
- `#define JAS_FIX_ABS(fix_t, fracbits, x)`
- `#define JAS_FIX_ISINT(fix_t, fracbits, x)`
- `#define JAS_FIX_SGN(fix_t, fracbits, x)`
- `#define JAS_FIX_CMP(fix_t, fracbits, x, y)`
- `#define JAS_FIX_LT(fix_t, fracbits, x, y)`
- `#define JAS_FIX_LTE(fix_t, fracbits, x, y)`
- `#define JAS_FIX_GT(fix_t, fracbits, x, y)`
- `#define JAS_FIX_GTE(fix_t, fracbits, x, y)`
- `#define JAS_FIX_ROUND(fix_t, fracbits, x)`
- `#define JAS_FIX_FLOOR(fix_t, fracbits, x)`

Typedefs

- `typedef int_least64_t jas_fix_t`

16.11.1 Detailed Description

JasPer Fixed-Point Number Class.

16.12 jas_fix.h

[Go to the documentation of this file.](#)

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia

```

```

00015 *
00016 * All rights reserved.
00017 *
00018 * Permission is hereby granted, free of charge, to any person (the
00019 * "User") obtaining a copy of this software and associated documentation
00020 * files (the "Software"), to deal in the Software without restriction,
00021 * including without limitation the rights to use, copy, modify, merge,
00022 * publish, distribute, and/or sell copies of the Software, and to permit
00023 * persons to whom the Software is furnished to do so, subject to the
00024 * following conditions:
00025 *
00026 * 1. The above copyright notices and this permission notice (which
00027 * includes the disclaimer below) shall be included in all copies or
00028 * substantial portions of the Software.
00029 *
00030 * 2. The name of a copyright holder shall not be used to endorse or
00031 * promote products derived from the Software without specific prior
00032 * written permission.
00033 *
00034 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060 *
00061 * __END_OF_JASPER_LICENSE__
00062 */
00063
00064
00065 #ifndef JAS_FIX_H
00066 #define JAS_FIX_H
00067
00068 /*****
00069 * Includes.
00070 */
00071
00072 /* The configuration header file should be included first. */
00073 #include <jasper/jas_config.h> /* IWYU pragma: keep */
00074
00075 #include <jasper/jas_types.h>
00076
00077 #ifdef __cplusplus
00078 extern "C" {
00079 #endif
00080
00081 /*****
00082 * Types.
00083 */
00084
00085 #if defined(JAS_ENABLE_32BIT)
00086 typedef int_least32_t jas_fix_t;
00087 #define PRIjas_fix PRIiLEAST32
00088 #else
00089 typedef int_least64_t jas_fix_t;
00090 #define PRIjas_fix PRIiLEAST64
00091 #endif
00092
00093 /*
00094 * Integral type used for double-precision fixed-point representations.

```

```

00107 */
00108 #if defined(JAS_ENABLE_32BIT)
00109 typedef int_least64_t jas_fix_big_t;
00110 #else
00111 #if defined(JAS_HAVE_INT128_T)
00112 typedef __int128_t jas_fix_big_t;
00113 #else
00114 typedef int_least64_t jas_fix_big_t;
00115 #endif
00116 #endif
00117
00118 /*****
00119  * Constants.
00120  *****/
00121
00122 #define JAS_FIX_ZERO(fix_t, fracbits) \
00123     JAS_INTTOFIX(fix_t, fracbits, 0)
00124
00125 #define JAS_FIX_ONE(fix_t, fracbits) \
00126     JAS_INTTOFIX(fix_t, fracbits, 1)
00127
00128 #define JAS_FIX_HALF(fix_t, fracbits) \
00129     (JAS_CAST(fix_t, 1) << ((fracbits) - 1))
00130
00131 /*****
00132  * Conversion operations.
00133  *****/
00134
00135 #define JAS_INTTOFIX(fix_t, fracbits, x) \
00136     (JAS_CAST(fix_t, x) << (fracbits))
00137
00138 #define JAS_FIXTOINT(fix_t, fracbits, x) \
00139     JAS_CAST(int, (x) >> (fracbits))
00140
00141 #define JAS_FIXTODBL(fix_t, fracbits, x) \
00142     (JAS_CAST(double, x) / JAS_FIX_ONE(fix_t, fracbits))
00143
00144 #define JAS_DBLTOFIX(fix_t, fracbits, x) \
00145     JAS_CAST(fix_t, ((x) * JAS_CAST(double, JAS_FIX_ONE(fix_t, fracbits))))
00146
00147 /*****
00148  * Basic arithmetic operations.
00149  * All other arithmetic operations are synthesized from these basic operations.
00150  * There are three macros for each type of arithmetic operation.
00151  * One macro always performs overflow/underflow checking, one never performs
00152  * overflow/underflow checking, and one is generic with its behavior
00153  * depending on compile-time flags.
00154  * Only the generic macros should be invoked directly by application code.
00155  *****/
00156
00157 #if !defined(DEBUG_OVERFLOW)
00158 #define JAS_FIX_ADD                JAS_FIX_ADD_FAST
00159 #else
00160 #define JAS_FIX_ADD                JAS_FIX_ADD_OFLOW
00161 #endif
00162
00163 #define JAS_FIX_ADD_FAST(fix_t, fracbits, x, y) ((x) + (y))
00164
00165 #define JAS_FIX_ADD_OFLOW(fix_t, fracbits, x, y) \
00166     ((x) >= 0) ? \
00167         (((y) >= 0) ? ((x) + (y) >= 0 || JAS_FIX_OFLOW(), (x) + (y)) : \
00168             ((x) + (y))) : \
00169         (((y) >= 0) ? ((x) + (y)) : ((x) + (y) < 0 || JAS_FIX_OFLOW(), \
00170             (x) + (y)))
00171
00172 #if !defined(DEBUG_OVERFLOW)
00173 #define JAS_FIX_MUL                JAS_FIX_MUL_FAST
00174 #else
00175 #define JAS_FIX_MUL                JAS_FIX_MUL_OFLOW
00176 #endif
00177
00178 #define JAS_FIX_MUL_FAST(fix_t, fracbits, bigfix_t, x, y) \
00179     JAS_CAST(fix_t, (JAS_CAST(bigfix_t, x) * JAS_CAST(bigfix_t, y)) >> \
00180         (fracbits))
00181
00182 #define JAS_FIX_MUL_OFLOW(fix_t, fracbits, bigfix_t, x, y) \
00183     ((JAS_CAST(bigfix_t, x) * JAS_CAST(bigfix_t, y)) >> (fracbits)) == \
00184     JAS_CAST(fix_t, (JAS_CAST(bigfix_t, x) * JAS_CAST(bigfix_t, y)) >> \
00185         (fracbits))) ? \
00186     JAS_CAST(fix_t, (JAS_CAST(bigfix_t, x) * JAS_CAST(bigfix_t, y)) >> \
00187         (fracbits))) : JAS_FIX_OFLOW()

```

```

00203
00205 #if !defined(DEBUG_OVERFLOW)
00206 #define JAS_FIX_MULBYINT          JAS_FIX_MULBYINT_FAST
00207 #else
00208 #define JAS_FIX_MULBYINT          JAS_FIX_MULBYINT_OFLOW
00209 #endif
00210
00213 #define JAS_FIX_MULBYINT_FAST(fix_t, fracbits, x, y) \
00214     JAS_CAST(fix_t, ((x) * (y)))
00215
00218 #define JAS_FIX_MULBYINT_OFLOW(fix_t, fracbits, x, y) \
00219     JAS_FIX_MULBYINT_FAST(fix_t, fracbits, x, y)
00220
00222 #if !defined(DEBUG_OVERFLOW)
00223 #define JAS_FIX_DIV          JAS_FIX_DIV_FAST
00224 #else
00225 #define JAS_FIX_DIV          JAS_FIX_DIV_UFLOW
00226 #endif
00227
00230 #define JAS_FIX_DIV_FAST(fix_t, fracbits, bigfix_t, x, y) \
00231     JAS_CAST(fix_t, (JAS_CAST(bigfix_t, x) « (fracbits)) / (y))
00232
00235 #define JAS_FIX_DIV_UFLOW(fix_t, fracbits, bigfix_t, x, y) \
00236     JAS_FIX_DIV_FAST(fix_t, fracbits, bigfix_t, x, y)
00237
00239 #if !defined(DEBUG_OVERFLOW)
00240 #define JAS_FIX_NEG          JAS_FIX_NEG_FAST
00241 #else
00242 #define JAS_FIX_NEG          JAS_FIX_NEG_OFLOW
00243 #endif
00244
00246 #define JAS_FIX_NEG_FAST(fix_t, fracbits, x) \
00247     (-x)
00248
00250 /* Yes, overflow is actually possible for two's complement representations,
00251    although highly unlikely to occur. */
00252 #define JAS_FIX_NEG_OFLOW(fix_t, fracbits, x) \
00253     (((x) < 0) ? (-x) > 0 || JAS_FIX_OFLOW(), -(x)) : -(x))
00254
00256 #if !defined(DEBUG_OVERFLOW)
00257 #define JAS_FIX_AS_L          JAS_FIX_AS_L_FAST
00258 #else
00259 #define JAS_FIX_AS_L          JAS_FIX_AS_L_OFLOW
00260 #endif
00261
00264 #define JAS_FIX_AS_L_FAST(fix_t, fracbits, x, n) \
00265     ((x) « (n))
00266
00269 #define JAS_FIX_AS_L_OFLOW(fix_t, fracbits, x, n) \
00270     (((x) « (n)) » (n)) == (x) || JAS_FIX_OFLOW(), (x) « (n))
00271
00273 #if !defined(DEBUG_OVERFLOW)
00274 #define JAS_FIX_AS_R          JAS_FIX_AS_R_FAST
00275 #else
00276 #define JAS_FIX_AS_R          JAS_FIX_AS_R_UFLOW
00277 #endif
00278
00281 #define JAS_FIX_AS_R_FAST(fix_t, fracbits, x, n) \
00282     ((x) » (n))
00283
00286 #define JAS_FIX_AS_R_UFLOW(fix_t, fracbits, x, n) \
00287     JAS_FIX_AS_R_FAST(fix_t, fracbits, x, n)
00288
00289 /*****
00290 * Other basic arithmetic operations.
00291 *****/
00292
00294 #define JAS_FIX_SUB(fix_t, fracbits, x, y) \
00295     JAS_FIX_ADD(fix_t, fracbits, x, JAS_FIX_NEG(fix_t, fracbits, y))
00296
00298 #define JAS_FIX_PLUSEQ(fix_t, fracbits, x, y) \
00299     ((x) = JAS_FIX_ADD(fix_t, fracbits, x, y))
00300
00302 #define JAS_FIX_MINUSEQ(fix_t, fracbits, x, y) \
00303     ((x) = JAS_FIX_SUB(fix_t, fracbits, x, y))
00304
00306 #define JAS_FIX_MULEQ(fix_t, fracbits, bigfix_t, x, y) \
00307     ((x) = JAS_FIX_MUL(fix_t, fracbits, bigfix_t, x, y))
00308
00309 /*****
00310 * Miscellaneous operations.

```

```

00311 \*****/
00312
00314 #define JAS_FIX_ABS(fix_t, fracbits, x) \
00315     ((x) >= 0) ? (x) : (JAS_FIX_NEG(fix_t, fracbits, x))
00316
00318 #define JAS_FIX_ISINT(fix_t, fracbits, x) \
00319     (JAS_FIX_FLOOR(fix_t, fracbits, x) == (x))
00320
00322 #define JAS_FIX_SGN(fix_t, fracbits, x) \
00323     ((x) >= 0 ? 1 : (-1))
00324
00325 /*****\
00326 * Relational operations.
00327 \*****/
00328
00330 #define JAS_FIX_CMP(fix_t, fracbits, x, y) \
00331     ((x) > (y) ? 1 : ((x) == (y)) ? 0 : (-1))
00332
00334 #define JAS_FIX_LT(fix_t, fracbits, x, y) \
00335     ((x) < (y))
00336
00338 #define JAS_FIX_LTE(fix_t, fracbits, x, y) \
00339     ((x) <= (y))
00340
00342 #define JAS_FIX_GT(fix_t, fracbits, x, y) \
00343     ((x) > (y))
00344
00346 #define JAS_FIX_GTE(fix_t, fracbits, x, y) \
00347     ((x) >= (y))
00348
00349 /*****\
00350 * Rounding functions.
00351 \*****/
00352
00354 #define JAS_FIX_ROUND(fix_t, fracbits, x) \
00355     (((x) < 0) ? JAS_FIX_FLOOR(fix_t, fracbits, JAS_FIX_ADD(fix_t, fracbits, \
00356     (x), JAS_FIX_HALF(fix_t, fracbits))) : \
00357     JAS_FIX_NEG(fix_t, fracbits, JAS_FIX_FLOOR(fix_t, fracbits, \
00358     JAS_FIX_ADD(fix_t, fracbits, (-x), JAS_FIX_HALF(fix_t, fracbits)))))
00359
00362 #define JAS_FIX_FLOOR(fix_t, fracbits, x) \
00363     ((x) & ~(JAS_FIX_ONE(fix_t, fracbits) - 1))
00364
00365 /*****\
00366 * The below macros are for internal library use only. Do not invoke them
00367 * directly in application code.
00368 \*****/
00369
00370 /* Handle overflow. */
00371 #define JAS_FIX_OFLOW() \
00372     jas_logerrorf("overflow error: file %s, line %d\n", __FILE__, __LINE__)
00373
00374 /* Handle underflow. */
00375 #define JAS_FIX_UFLOW() \
00376     jas_logerrorf("underflow error: file %s, line %d\n", __FILE__, __LINE__)
00377
00378 /*****\
00379 *
00380 \*****/
00381
00382 JAS_ATTRIBUTE_DISABLE_UBSAN
00383 static inline jas_fix_t jas_fix_asl(jas_fix_t x, unsigned n)
00384 {
00385     return JAS_FIX_AS_L(jas_fix_t, 0, x, n);
00386 }
00387
00388 JAS_ATTRIBUTE_DISABLE_UBSAN
00389 static inline jas_fix_t jas_fix_asr(jas_fix_t x, unsigned n)
00390 {
00391     return JAS_FIX_AS_R(jas_fix_t, 0, x, n);
00392 }
00393
00397
00398 #ifdef __cplusplus
00399 }
00400 #endif
00401
00402 #endif

```

16.13 jas_getopt.h File Reference

Command Line Option Parsing Code.

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

Classes

- struct [jas_opt_t](#)
Command line option type.

Macros

- #define [JAS_GETOPT_EOF](#) (-1)
- #define [JAS_GETOPT_ERR](#) '?'
- #define [JAS_OPT_HASARG](#) 0x01 /* option has argument */

Functions

- JAS_EXPORT int [jas_getopt](#) (int argc, char **argv, const [jas_opt_t](#) *opts)
Get the next option.

Variables

- JAS_EXPORT int [jas_optind](#)
The current option index.
- JAS_EXPORT const char * [jas_optarg](#)
The current option argument.
- JAS_EXPORT int [jas_opterr](#)
The debug level.

16.13.1 Detailed Description

Command Line Option Parsing Code.

16.14 jas_getopt.h

[Go to the documentation of this file.](#)

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * Jasper License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00064
00065 #ifndef JAS_GETOPT_H
00066 #define JAS_GETOPT_H
00067
00068 #ifdef __cplusplus
00069 extern "C" {
00070 #endif
00071
00072 /* The configuration header file should be included first. */
00073 #include <jasper/jas_config.h>
00074
00075
00076

```

```

00084 /*****
00085  * Constants.
00086  *****/
00087
00089 #define JAS_GETOPT_EOF    (-1)
00091 #define JAS_GETOPT_ERR    '?'
00092
00094 #define JAS_OPT_HASARG    0x01    /* option has argument */
00095
00096 /*****
00097  * Types.
00098  *****/
00099
00104 typedef struct {
00105
00106     /* The unique identifier for this option. */
00107     int id;
00108
00109     /* The name of this option. */
00110     const char *name;
00111
00112     /* option flags. */
00113     int flags;
00114
00115 } jas_opt_t;
00116
00117 /*****
00118  * External data.
00119  *****/
00120
00125 JAS_EXPORT
00126 extern int jas_optind;
00127
00132 JAS_EXPORT
00133 extern const char *jas_optarg;
00134
00139 JAS_EXPORT
00140 extern int jas_opterr;
00141
00142 /*****
00143  * Prototypes.
00144  *****/
00145
00157 JAS_EXPORT
00158 int jas_getopt(int argc, char **argv, const jas_opt_t *opts);
00159
00163
00164 #ifdef __cplusplus
00165 }
00166 #endif
00167
00168 #endif

```

16.15 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>

```

Functions

- JAS_EXPORT jas_iccprof_t * [jas_iccprof_load](#) (jas_stream_t *in)
Read an ICC profile from a stream.

- JAS_EXPORT int [jas_iccprof_save](#) (jas_iccprof_t *prof, [jas_stream_t](#) *out)
Write an ICC profile to a stream.
- JAS_EXPORT void [jas_iccprof_destroy](#) (jas_iccprof_t *prof)
Destroy an ICC profile.
- JAS_ATTRIBUTE_PURE JAS_EXPORT jas_iccattrval_t * [jas_iccprof_getattr](#) (const jas_iccprof_t *prof, jas_iccattrname_t name)
Get an attribute of an ICC profile.
- JAS_EXPORT int [jas_iccprof_setattr](#) (jas_iccprof_t *prof, jas_iccattrname_t name, jas_iccattrval_t *val)
Set an attribute of an ICC profile.
- JAS_EXPORT void [jas_iccprof_dump](#) (const jas_iccprof_t *prof, FILE *out)
Dump an ICC profile to a stream in human-readable format for debugging purposes.
- JAS_EXPORT jas_iccprof_t * [jas_iccprof_copy](#) (const jas_iccprof_t *prof)
Create a copy of an ICC profile.
- JAS_EXPORT int [jas_iccprof_gethdr](#) (const jas_iccprof_t *prof, jas_icchdr_t *hdr)
Get the header for an ICC profile.
- JAS_EXPORT int [jas_iccprof_sethdr](#) (jas_iccprof_t *prof, const jas_icchdr_t *hdr)
Set the header for an ICC profile.
- JAS_EXPORT void [jas_iccattrval_destroy](#) (jas_iccattrval_t *attrval)
Destroy an ICC profile attribute.
- JAS_EXPORT int [jas_iccattrval_allowmodify](#) (jas_iccattrval_t **attrval)
TODO/FIXME.
- JAS_EXPORT jas_iccattrval_t * [jas_iccattrval_clone](#) (jas_iccattrval_t *attrval)
Create a copy of an ICC profile attribute.
- JAS_EXPORT jas_iccattrval_t * [jas_iccattrval_create](#) (jas_iccuint32_t type)
Create an ICC profile attribute.
- JAS_EXPORT void [jas_iccattrtab_dump](#) (const jas_iccattrtab_t *attrtab, FILE *out)
Dump an ICC profile attribute to a stream in human-readable format for debugging purposes.
- JAS_EXPORT jas_iccprof_t * [jas_iccprof_createfrombuf](#) (const jas_uchar *buf, unsigned len)
Create an ICC profile from a buffer in memory.
- JAS_EXPORT jas_iccprof_t * [jas_iccprof_createfromclrspc](#) (unsigned clrspc)
Create an ICC profile from a color space.

16.15.1 Detailed Description

ICC Profile.

16.16 jas_icc.h

[Go to the documentation of this file.](#)

```

00001 /*
00002  * Copyright (c) 2002-2003 Michael David Adams.
00003  * All rights reserved.
00004  */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007  *
00008  * JasPer License Version 2.0
00009  */

```

```

00010 * Copyright (c) 2001-2006 Michael David Adams
00011 * Copyright (c) 1999-2000 Image Power, Inc.
00012 * Copyright (c) 1999-2000 The University of British Columbia
00013 *
00014 * All rights reserved.
00015 *
00016 * Permission is hereby granted, free of charge, to any person (the
00017 * "User") obtaining a copy of this software and associated documentation
00018 * files (the "Software"), to deal in the Software without restriction,
00019 * including without limitation the rights to use, copy, modify, merge,
00020 * publish, distribute, and/or sell copies of the Software, and to permit
00021 * persons to whom the Software is furnished to do so, subject to the
00022 * following conditions:
00023 *
00024 * 1. The above copyright notices and this permission notice (which
00025 * includes the disclaimer below) shall be included in all copies or
00026 * substantial portions of the Software.
00027 *
00028 * 2. The name of a copyright holder shall not be used to endorse or
00029 * promote products derived from the Software without specific prior
00030 * written permission.
00031 *
00032 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00054 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058 *
00059 * __END_OF_JASPER_LICENSE__
00060 */
00061
00066
00067 #ifndef JAS_ICC_H
00068 #define JAS_ICC_H
00069
00070 /* The configuration header file should be included first. */
00071 #include <jasper/jas_config.h>
00072
00073 #include <jasper/jas_types.h>
00074 #include <jasper/jas_stream.h>
00075
00076 #include <stdio.h>
00077
00078 #ifdef __cplusplus
00079 extern "C" {
00080 #endif
00081
00086
00087 /* Profile file signature. */
00088 #define JAS_ICC_MAGIC 0x61637370
00089
00090 #define JAS_ICC_HDRLEN 128
00091
00092 /* Profile/device class signatures. */
00093 #define JAS_ICC_CLAS_IN 0x73636e72 /* input device */
00094 #define JAS_ICC_CLAS_DPY 0x6d6e7472 /* display device */
00095 #define JAS_ICC_CLAS_OUT 0x70727472 /* output device */
00096 #define JAS_ICC_CLAS_LNK 0x6c696e6b /* device link */
00097 #define JAS_ICC_CLAS_CNV 0x73706163 /* color space conversion */
00098 #define JAS_ICC_CLAS_ABS 0x61627374 /* abstract */

```

```
00099 #define JAS_ICC_CLAS_NAM      0x6e6d636c /* named color */
00100
00101 /* Color space signatures. */
00102 #define JAS_ICC_COLORSPC_XYZ    0x58595a20 /* XYZ */
00103 #define JAS_ICC_COLORSPC_LAB    0x4c616220 /* LAB */
00104 #define JAS_ICC_COLORSPC_LUV    0x4c757620 /* LUV */
00105 #define JAS_ICC_COLORSPC_YCBCR  0x59436272 /* YCbCr */
00106 #define JAS_ICC_COLORSPC_YXY    0x59787920 /* Yxy */
00107 #define JAS_ICC_COLORSPC_RGB    0x52474220 /* RGB */
00108 #define JAS_ICC_COLORSPC_GRAY   0x47524159 /* Gray */
00109 #define JAS_ICC_COLORSPC_HSV    0x48535620 /* HSV */
00110 #define JAS_ICC_COLORSPC_HLS    0x484c5320 /* HLS */
00111 #define JAS_ICC_COLORSPC_CMYK   0x434d594b /* CMYK */
00112 #define JAS_ICC_COLORSPC_CMY    0x434d5920 /* CMY */
00113 #define JAS_ICC_COLORSPC_2      0x32434c52 /* 2 channel color */
00114 #define JAS_ICC_COLORSPC_3      0x33434c52 /* 3 channel color */
00115 #define JAS_ICC_COLORSPC_4      0x34434c52 /* 4 channel color */
00116 #define JAS_ICC_COLORSPC_5      0x35434c52 /* 5 channel color */
00117 #define JAS_ICC_COLORSPC_6      0x36434c52 /* 6 channel color */
00118 #define JAS_ICC_COLORSPC_7      0x37434c52 /* 7 channel color */
00119 #define JAS_ICC_COLORSPC_8      0x38434c52 /* 8 channel color */
00120 #define JAS_ICC_COLORSPC_9      0x39434c52 /* 9 channel color */
00121 #define JAS_ICC_COLORSPC_10     0x41434c52 /* 10 channel color */
00122 #define JAS_ICC_COLORSPC_11     0x42434c52 /* 11 channel color */
00123 #define JAS_ICC_COLORSPC_12     0x43434c52 /* 12 channel color */
00124 #define JAS_ICC_COLORSPC_13     0x44434c52 /* 13 channel color */
00125 #define JAS_ICC_COLORSPC_14     0x45434c52 /* 14 channel color */
00126 #define JAS_ICC_COLORSPC_15     0x46434c52 /* 15 channel color */
00127
00128 /* Profile connection color space (PCS) signatures. */
00129 #define JAS_ICC_REFCOLORSPC_XYZ  0x58595a20 /* CIE XYZ */
00130 #define JAS_ICC_REFCOLORSPC_LAB  0x4c616220 /* CIE Lab */
00131
00132 /* Primary platform signatures. */
00133 #define JAS_ICC_PLATFORM_APPL    0x4150504c /* Apple Computer */
00134 #define JAS_ICC_PLATFORM_MSFT    0x4d534654 /* Microsoft */
00135 #define JAS_ICC_PLATFORM_SGI     0x53474920 /* Silicon Graphics */
00136 #define JAS_ICC_PLATFORM_SUNW    0x53554e57 /* Sun Microsystems */
00137 #define JAS_ICC_PLATFORM_TGNT    0x54474e54 /* Taligent */
00138
00139 /* Profile flags. */
00140 #define JAS_ICC_FLAGS_EMBED      0x01 /* embedded */
00141 #define JAS_ICC_FLAGS_NOSEP      0x02 /* no separate use */
00142
00143 /* Attributes. */
00144 #define JAS_ICC_ATTR_TRANS       0x01 /* transparent */
00145 #define JAS_ICC_ATTR_MATTE       0x02 /* matte */
00146
00147 /* Rendering intents. */
00148 #define JAS_ICC_INTENT_PER        0 /* perceptual */
00149 #define JAS_ICC_INTENT_REL        1 /* relative colorimetric */
00150 #define JAS_ICC_INTENT_SAT        2 /* saturation */
00151 #define JAS_ICC_INTENT_ABS        3 /* absolute colorimetric */
00152
00153 /* Tag signatures. */
00154 #define JAS_ICC_TAG_ATOB0        0x41324230 /* */
00155 #define JAS_ICC_TAG_ATOB1        0x41324231 /* */
00156 #define JAS_ICC_TAG_ATOB2        0x41324232 /* */
00157 #define JAS_ICC_TAG_BLUMATCOL    0x6258595a /* */
00158 #define JAS_ICC_TAG_BLUTRC       0x62545243 /* */
00159 #define JAS_ICC_TAG_BTOA0        0x42324130 /* */
00160 #define JAS_ICC_TAG_BTOA1        0x42324131 /* */
00161 #define JAS_ICC_TAG_BTOA2        0x42324132 /* */
00162 #define JAS_ICC_TAG_CALTIME       0x63616c74 /* */
00163 #define JAS_ICC_TAG_CHARTARGET    0x74617267 /* */
00164 #define JAS_ICC_TAG_CPYRT        0x63707274 /* */
00165 #define JAS_ICC_TAG_CRDINFO       0x63726469 /* */
00166 #define JAS_ICC_TAG_DEVMAKERDESC  0x646d6e64 /* */
00167 #define JAS_ICC_TAG_DEVMODELDESC  0x646d6464 /* */
00168 #define JAS_ICC_TAG_DEVSET        0x64657673 /* */
00169 #define JAS_ICC_TAG_GAMUT         0x67616d74 /* */
00170 #define JAS_ICC_TAG_GRYTRC       0x6b545243 /* */
00171 #define JAS_ICC_TAG_GRNMATCOL    0x6758595a /* */
00172 #define JAS_ICC_TAG_GRNTRC       0x67545243 /* */
00173 #define JAS_ICC_TAG_LUM           0x6c756d69 /* */
00174 #define JAS_ICC_TAG_MEASURE       0x6d656173 /* */
00175 #define JAS_ICC_TAG_MEDIABLKPT    0x626b7074 /* */
00176 #define JAS_ICC_TAG_MEDIABHIPT    0x77747074 /* */
00177 #define JAS_ICC_TAG_NAMCOLR       0x6e636f63 /* */
00178 #define JAS_ICC_TAG_NAMCOLR2      0x6e636c32 /* */
00179 #define JAS_ICC_TAG_OUTRESP       0x72657370 /* */
```

```

00180 #define JAS_ICC_TAG_PREVIEW0      0x70726530 /* */
00181 #define JAS_ICC_TAG_PREVIEW1      0x70726531 /* */
00182 #define JAS_ICC_TAG_PREVIEW2      0x70726532 /* */
00183 #define JAS_ICC_TAG_PROFDESC      0x64657363 /* */
00184 #define JAS_ICC_TAG_PROFSEQDESC    0x70736571 /* */
00185 #define JAS_ICC_TAG_PSDCRD0      0x70736430 /* */
00186 #define JAS_ICC_TAG_PSCRDD1      0x70736431 /* */
00187 #define JAS_ICC_TAG_PSCRDD2      0x70736432 /* */
00188 #define JAS_ICC_TAG_PSCRDD3      0x70736433 /* */
00189 #define JAS_ICC_TAG_PS2CSA      0x70733273 /* */
00190 #define JAS_ICC_TAG_PS2RENINTENT    0x70733269 /* */
00191 #define JAS_ICC_TAG_REDMATCOL      0x7258595a /* */
00192 #define JAS_ICC_TAG_REDTRC      0x72545243 /* */
00193 #define JAS_ICC_TAG_SCRNGDES      0x73637264 /* */
00194 #define JAS_ICC_TAG_SCRNG      0x7363726e /* */
00195 #define JAS_ICC_TAG_TECH      0x74656368 /* */
00196 #define JAS_ICC_TAG_UCRBG      0x62666420 /* */
00197 #define JAS_ICC_TAG_VIEWCONDESC    0x76756564 /* */
00198 #define JAS_ICC_TAG_VIEWCOND      0x76696577 /* */
00199
00200 /* Type signatures. */
00201 #define JAS_ICC_TYPE_CRDINFO      0x63726469 /* CRD information */
00202 #define JAS_ICC_TYPE_CURV      0x63757276 /* curve */
00203 #define JAS_ICC_TYPE_DATA      0x64617461 /* data */
00204 #define JAS_ICC_TYPE_TIME      0x6474696d /* date/time */
00205 #define JAS_ICC_TYPE_DEVSET      0x64657673 /* device settings */
00206 #define JAS_ICC_TYPE_LUT16      0x6d667432 /* */
00207 #define JAS_ICC_TYPE_LUT8      0x6d667431 /* */
00208 #define JAS_ICC_TYPE_MEASURE      0x6d656173 /* */
00209 #define JAS_ICC_TYPE_NAMCOLR      0x6e63666c /* */
00210 #define JAS_ICC_TYPE_NAMCOLR2      0x6e6366c32 /* */
00211 #define JAS_ICC_TYPE_PROFSEQDESC    0x70736571 /* profile sequence description */
00212 #define JAS_ICC_TYPE_RESPCURVSET16 0x72637332 /* response curve set 16 */
00213 #define JAS_ICC_TYPE_SF32      0x73663332 /* signed 32-bit fixed-point */
00214 #define JAS_ICC_TYPE_SCRNG      0x7363726e /* screening */
00215 #define JAS_ICC_TYPE_SIG      0x73696720 /* signature */
00216 #define JAS_ICC_TYPE_TXTDESC      0x64657363 /* text description */
00217 #define JAS_ICC_TYPE_TXT      0x74657874 /* text */
00218 #define JAS_ICC_TYPE_UF32      0x75663332 /* unsigned 32-bit fixed-point */
00219 #define JAS_ICC_TYPE_UCRBG      0x62666420 /* */
00220 #define JAS_ICC_TYPE_UI16      0x75693136 /* */
00221 #define JAS_ICC_TYPE_UI32      0x75693332 /* */
00222 #define JAS_ICC_TYPE_UI8      0x75693038 /* */
00223 #define JAS_ICC_TYPE_UI64      0x75693634 /* */
00224 #define JAS_ICC_TYPE_VIEWCOND      0x76696577 /* */
00225 #define JAS_ICC_TYPE_XYZ      0x58595a20 /* XYZ */
00226
00227 typedef uint_fast8_t jas_iccuint8_t;
00228 typedef uint_fast16_t jas_iccuint16_t;
00229 typedef uint_fast32_t jas_iccuint32_t;
00230 typedef int_fast32_t jas_iccsint32_t;
00231 typedef int_fast32_t jas_iccs15fixed16_t;
00232 typedef uint_fast32_t jas_iccul6fixed16_t;
00233 typedef uint_fast64_t jas_iccuint64_t;
00234 typedef uint_fast32_t jas_iccsig_t;
00235
00236 typedef jas_iccsig_t jas_icctagsig_t;
00237 typedef jas_iccsig_t jas_icctagtype_t;
00238 typedef jas_iccsig_t jas_iccattrname_t;
00239
00240 /* Date/time type. */
00241 typedef struct {
00242     jas_iccuint16_t year;
00243     jas_iccuint16_t month;
00244     jas_iccuint16_t day;
00245     jas_iccuint16_t hour;
00246     jas_iccuint16_t min;
00247     jas_iccuint16_t sec;
00248 } jas_icctime_t;
00249
00250 /* XYZ type. */
00251 typedef struct {
00252     jas_iccs15fixed16_t x;
00253     jas_iccs15fixed16_t y;
00254     jas_iccs15fixed16_t z;
00255 } jas_icxyz_t;
00256
00257 /* Curve type. */
00258 typedef struct {
00259     jas_iccuint32_t numents;
00260     jas_iccuint16_t *ents;

```

```

00261 } jas_icccurv_t;
00262
00263 /* Text description type. */
00264 typedef struct {
00265     jas_iccuint32_t asclen;
00266     char *ascdata; /* ASCII invariant description */
00267     jas_iccuint32_t uclangcode; /* Unicode language code */
00268     jas_iccuint32_t uclen; /* Unicode localizable description count */
00269     jas_uchar *ucdata; /* Unicode localizable description */
00270     jas_iccuint16_t sccode; /* ScriptCode code */
00271     jas_iccuint8_t maclen; /* Localizable Macintosh description count */
00272     jas_uchar macdata[69]; /* Localizable Macintosh description */
00273 } jas_icctxtdesc_t;
00274
00275 /* Text type. */
00276 typedef struct {
00277     char *string; /* ASCII character string */
00278 } jas_icctxt_t;
00279
00280 typedef struct {
00281     jas_iccuint8_t numinchans;
00282     jas_iccuint8_t numoutchans;
00283     jas_iccsint32_t e[3][3];
00284     jas_iccuint8_t clutlen;
00285     jas_iccuint8_t *clut;
00286     jas_iccuint16_t numintabents;
00287     jas_iccuint8_t **intabs;
00288     jas_iccuint8_t *intabsbuf;
00289     jas_iccuint16_t numouttabents;
00290     jas_iccuint8_t **outtabs;
00291     jas_iccuint8_t *outtabsbuf;
00292 } jas_icclut8_t;
00293
00294 typedef struct {
00295     jas_iccuint8_t numinchans;
00296     jas_iccuint8_t numoutchans;
00297     jas_iccsint32_t e[3][3];
00298     jas_iccuint8_t clutlen;
00299     jas_iccuint16_t *clut;
00300     jas_iccuint16_t numintabents;
00301     jas_iccuint16_t **intabs;
00302     jas_iccuint16_t *intabsbuf;
00303     jas_iccuint16_t numouttabents;
00304     jas_iccuint16_t **outtabs;
00305     jas_iccuint16_t *outtabsbuf;
00306 } jas_icclut16_t;
00307
00308 struct jas_iccattrval_s;
00309
00310 typedef struct {
00311     void (*destroy)(struct jas_iccattrval_s *);
00312     int (*copy)(struct jas_iccattrval_s *, const struct jas_iccattrval_s *);
00313     int (*input)(struct jas_iccattrval_s *, jas_stream_t *, unsigned);
00314     /*#ifdef JAS_ENABLE_ENCODER
00315     int (*output)(struct jas_iccattrval_s *, jas_stream_t *);
00316     /*#endif
00317     unsigned (*getsize)(const struct jas_iccattrval_s *);
00318     void (*dump)(const struct jas_iccattrval_s *, FILE *);
00319 } jas_iccattrvalops_t;
00320
00321 /* Attribute value type (type and value information). */
00322 typedef struct jas_iccattrval_s {
00323     unsigned refcnt; /* reference count */
00324     jas_iccsig_t type; /* type */
00325     const jas_iccattrvalops_t *ops; /* type-dependent operations */
00326     union {
00327         jas_iccxyz_t xyz;
00328         jas_icccurv_t curv;
00329         jas_icctxtdesc_t txtdesc;
00330         jas_icctxt_t txt;
00331         jas_icclut8_t lut8;
00332         jas_icclut16_t lut16;
00333     } data; /* value */
00334 } jas_iccattrval_t;
00335
00336 /* Header type. */
00337 typedef struct {
00338     jas_iccuint32_t size; /* profile size */
00339     jas_iccsig_t cmmtype; /* CMM type signature */
00340     jas_iccuint32_t version; /* profile version */
00341     jas_iccsig_t clas; /* profile/device class signature */

```

```

00342     jas_iccsig_t colorspace; /* color space of data */
00343     jas_iccsig_t refcolorspace; /* profile connection space */
00344     jas_icctime_t ctime; /* creation time */
00345     jas_iccsig_t magic; /* profile file signature */
00346     jas_iccsig_t platform; /* primary platform */
00347     jas_iccuint32_t flags; /* profile flags */
00348     jas_iccsig_t maker; /* device manufacturer signature */
00349     jas_iccsig_t model; /* device model signature */
00350     jas_iccuint64_t attr; /* device setup attributes */
00351     jas_iccsig_t intent; /* rendering intent */
00352     jas_iccxyz_t illum; /* illuminant */
00353     jas_iccsig_t creator; /* profile creator signature */
00354 } jas_icchdr_t;
00355
00356 typedef struct {
00357     jas_iccsig_t name;
00358     jas_iccattrval_t *val;
00359 } jas_iccattr_t;
00360
00361 typedef struct {
00362     unsigned numattrs;
00363     unsigned maxattrs;
00364     jas_iccattr_t *attrs;
00365 } jas_iccattrtab_t;
00366
00367 typedef struct jas_icctagtabent_s {
00368     jas_iccuint32_t tag;
00369     jas_iccuint32_t off;
00370     jas_iccuint32_t len;
00371     void *data;
00372     struct jas_icctagtabent_s *first;
00373 } jas_icctagtabent_t;
00374
00375 typedef struct {
00376     jas_iccuint32_t numents;
00377     jas_icctagtabent_t *ents;
00378 } jas_icctagtab_t;
00379
00380 /* ICC profile type. */
00381 typedef struct {
00382     jas_icchdr_t hdr;
00383     jas_icctagtab_t tagtab;
00384     jas_iccattrtab_t attrtab;
00385 } jas_iccprof_t;
00386
00387 typedef struct {
00388     jas_iccuint32_t type;
00389     jas_iccattrvalops_t ops;
00390 } jas_iccattrvalinfo_t;
00391
00396 JAS_EXPORT
00397 jas_iccprof_t *jas_iccprof_load(jas_stream_t *in);
00398
00403 JAS_EXPORT
00404 int jas_iccprof_save(jas_iccprof_t *prof, jas_stream_t *out);
00405
00410 JAS_EXPORT
00411 void jas_iccprof_destroy(jas_iccprof_t *prof);
00412
00417 JAS_ATTRIBUTE_PURE
00418 JAS_EXPORT
00419 jas_iccattrval_t *jas_iccprof_getattr(const jas_iccprof_t *prof,
00420     jas_iccattrname_t name);
00421
00426 JAS_EXPORT
00427 int jas_iccprof_setattr(jas_iccprof_t *prof, jas_iccattrname_t name,
00428     jas_iccattrval_t *val);
00429
00435 JAS_EXPORT
00436 void jas_iccprof_dump(const jas_iccprof_t *prof, FILE *out);
00437
00442 JAS_EXPORT
00443 jas_iccprof_t *jas_iccprof_copy(const jas_iccprof_t *prof);
00444
00449 JAS_EXPORT
00450 int jas_iccprof_gethdr(const jas_iccprof_t *prof, jas_icchdr_t *hdr);
00451
00456 JAS_EXPORT
00457 int jas_iccprof_sethdr(jas_iccprof_t *prof, const jas_icchdr_t *hdr);
00458
00463 JAS_EXPORT

```



```

00464 void jas_iccattrval_destroy(jas_iccattrval_t *attrval);
00465
00469 JAS_EXPORT
00470 void jas_iccattrval_dump(const jas_iccattrval_t *attrval, FILE *out);
00471
00477 JAS_EXPORT
00478 int jas_iccattrval_allowmodify(jas_iccattrval_t **attrval);
00479
00484 JAS_EXPORT
00485 jas_iccattrval_t *jas_iccattrval_clone(jas_iccattrval_t *attrval);
00486
00491 JAS_EXPORT
00492 jas_iccattrval_t *jas_iccattrval_create(jas_iccuint32_t type);
00493
00499 JAS_EXPORT
00500 void jas_iccatrrtab_dump(const jas_iccatrrtab_t *attrtab, FILE *out);
00501
00506 JAS_EXPORT
00507 jas_iccprof_t *jas_iccprof_createfrombuf(const jas_uchar *buf, unsigned len);
00508
00513 JAS_EXPORT
00514 jas_iccprof_t *jas_iccprof_createfromclrspc(unsigned clrspc);
00515
00516 JAS_EXPORT
00517 extern const jas_uchar jas_iccprofdata_srgb[];
00518
00519 JAS_EXPORT
00520 extern const unsigned jas_iccprofdata_srgbllen;
00521
00522 JAS_EXPORT
00523 extern const jas_uchar jas_iccprofdata_sgray[];
00524
00525 JAS_EXPORT
00526 extern const unsigned jas_iccprofdata_sgraylen;
00527
00531
00532 #ifdef __cplusplus
00533 }
00534 #endif
00535
00536 #endif

```

16.17 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_fmtdinfo_t](#)
Image format information.

Macros

- `#define JAS_IMAGE_MAXFMTS 32`
The maximum number of image data formats supported.
- `#define jas_image_width(image)`
Get the width of the image in units of the image reference grid.
- `#define jas_image_height(image)`
Get the height of the image in units of the image reference grid.
- `#define jas_image_tlx(image)`
Get the x-coordinate of the top-left corner of the image bounding box on the reference grid.
- `#define jas_image_tly(image)`
Get the y-coordinate of the top-left corner of the image bounding box on the reference grid.
- `#define jas_image_brx(image)`
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)`
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)`
Get the number of image components.
- `#define jas_image_clrspc(image)`
Get the color model used by the image.
- `#define jas_image_setclrspc(image, clrspc)`
Set the color model for an image.
- `#define jas_image_cmptwidth(image, cmptno)`
Get the width of a component.
- `#define jas_image_cmptheight(image, cmptno)`
Get the height of a component.
- `#define jas_image_cmptsgnd(image, cmptno)`
Get the signedness of the sample data for a component.
- `#define jas_image_cmptprec(image, cmptno)`
Get the precision of the sample data for a component.
- `#define jas_image_cmptstep(image, cmptno)`
Get the horizontal subsampling factor for a component.
- `#define jas_image_cmptvstep(image, cmptno)`
Get the vertical subsampling factor for a component.
- `#define jas_image_cmpttlx(image, cmptno)`
Get the x-coordinate of the top-left corner of a component.
- `#define jas_image_cmpttly(image, cmptno)`
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_cmprofi(image)`
Get the color management profile of an image.
- `#define jas_image_setcmprofi(image, cmprofi)`
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_EXPORT [jas_image_t](#) * [jas_image_create](#) (unsigned numcmpts, const [jas_image_cmptparm_t](#) *cmptparms, [jas_clrspc_t](#) clrspc)
Create an image.
- JAS_EXPORT [jas_image_t](#) * [jas_image_create0](#) (void)
Create an "empty" image.
- JAS_EXPORT [jas_image_t](#) * [jas_image_copy](#) ([jas_image_t](#) *image)
Clone an image.
- JAS_EXPORT void [jas_image_destroy](#) ([jas_image_t](#) *image)
Deallocate any resources associated with an image.
- JAS_ATTRIBUTE_PURE JAS_EXPORT 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_EXPORT 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_EXPORT [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_EXPORT 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_EXPORT 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_EXPORT 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_EXPORT void [jas_image_delcmpt](#) ([jas_image_t](#) *image, unsigned cmptno)
Delete a component from an image.
- JAS_EXPORT int [jas_image_addcmpt](#) ([jas_image_t](#) *image, int cmptno, const [jas_image_cmptparm_t](#) *cmptparm)
Add a component to an image.
- JAS_EXPORT 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_EXPORT 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_EXPORT int [jas_image_readcmptsample](#) ([jas_image_t](#) *image, unsigned cmptno, unsigned x, unsigned y)

Read a component sample for an image.

- JAS_EXPORT 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.

- JAS_ATTRIBUTE_PURE JAS_EXPORT int [jas_image_getcmptbytype](#) (const [jas_image_t](#) *image, [jas_image_cmpttype_t](#) ctype)

Get an image component by its type.

- JAS_EXPORT void [jas_image_clearfmts](#) (void)

Clear the table of image formats.

- JAS_EXPORT const [jas_image_fmtinfo_t](#) * [jas_image_getfmtbyind](#) (int index)

Get a image format entry by its table index.

- JAS_EXPORT int [jas_image_getnumfmts](#) (void)

Get the number of image format table entries.

- JAS_EXPORT int [jas_image_setfmtenable](#) (int index, int enabled)

Get the number of image format table entries.

- JAS_EXPORT 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_EXPORT int [jas_image_strtofmt](#) (const char *s)

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

- JAS_ATTRIBUTE_CONST JAS_EXPORT const char * [jas_image_fmtostr](#) (int fmt)

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

- JAS_ATTRIBUTE_CONST JAS_EXPORT const [jas_image_fmtinfo_t](#) * [jas_image_lookupfmtbyid](#) (int id)

Lookup image format information by the format ID.

- JAS_ATTRIBUTE_PURE JAS_EXPORT const [jas_image_fmtinfo_t](#) * [jas_image_lookupfmtbyname](#) (const char *name)

Lookup image format information by the format name.

- JAS_ATTRIBUTE_PURE JAS_EXPORT int [jas_image_fmtfromname](#) (const char *filename)

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

- JAS_ATTRIBUTE_PURE JAS_EXPORT int [jas_image_getfmt](#) ([jas_stream_t](#) *in)

Get the format of image data in a stream.

- JAS_ATTRIBUTE_PURE JAS_EXPORT int [jas_image_ishomosamp](#) (const [jas_image_t](#) *image)

Test if the sampling of the image is homogeneous.

- JAS_EXPORT 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_EXPORT int [jas_image_writempt2](#) ([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_EXPORT 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_EXPORT [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_EXPORT int [jas_image_dump](#) ([jas_image_t](#) *image, FILE *out)

Dump the information for an image (for debugging).

16.17.1 Detailed Description

JasPer Image Class.

16.18 jas_image.h

[Go to the documentation of this file.](#)

```
00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2003 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00068
00069 #ifndef JAS_IMAGE_H
00070 #define JAS_IMAGE_H
```

```

00071
00072 /*****
00073  * Includes.
00074  *****/
00075
00076 /* The configuration header file should be included first. */
00077 #include <jasper/jas_config.h>
00078
00079 #include <jasper/jas_stream.h>
00080 #include <jasper/jas_types.h>
00081 #include <jasper/jas_seq.h> /* IWYU pragma: export */
00082 #include <jasper/jas_cm.h> /* IWYU pragma: export */
00083 #include <stdio.h>
00084
00085 #ifdef __cplusplus
00086 extern "C" {
00087 #endif
00088
00093
00094 /*****
00095  * Constants.
00096  *****/
00097
00098 /*
00099  * Miscellaneous constants.
00100  */
00101
00102 /* Basic units */
00103 #define JAS_IMAGE_KIBI (JAS_CAST(size_t, 1024))
00104 #define JAS_IMAGE_MEBI (JAS_IMAGE_KIBI * JAS_IMAGE_KIBI)
00105
00106 /* The threshold at which image data is no longer stored in memory. */
00107 #define JAS_IMAGE_INMEMTHRESH (256 * JAS_IMAGE_MEBI)
00108
00109 /*
00110  * Component types
00111  */
00112
00113 #define JAS_IMAGE_CT_UNKNOWN 0x10000
00114 #define JAS_IMAGE_CT_COLOR(n) ((n) & 0x7fff)
00115 #define JAS_IMAGE_CT_OPACITY 0x08000
00116
00117 #define JAS_IMAGE_CT_RGB_R 0
00118 #define JAS_IMAGE_CT_RGB_G 1
00119 #define JAS_IMAGE_CT_RGB_B 2
00120
00121 #define JAS_IMAGE_CT_YCBCR_Y 0
00122 #define JAS_IMAGE_CT_YCBCR_CB 1
00123 #define JAS_IMAGE_CT_YCBCR_CR 2
00124
00125 #define JAS_IMAGE_CT_GRAY_Y 0
00126
00127 /*****
00128  * Simple types.
00129  *****/
00130
00131 typedef int_fast32_t jas_image_coord_t;
00132 #define JAS_IMAGE_COORD_MAX INT_FAST32_MAX
00133 #define JAS_IMAGE_COORD_MIN INT_FAST32_MIN
00134
00135 typedef int_fast16_t jas_image_colorspc_t;
00136
00137 typedef int_fast32_t jas_image_cmpttype_t;
00138
00139 typedef int_fast16_t jas_image_smpltype_t;
00140
00141 /*****
00142  * Image class and supporting classes.
00143  *****/
00144
00145 typedef struct {
00146
00147     /* The x-coordinate of the top-left corner of the component. */
00148     jas_image_coord_t tlx_;
00149
00150     /* The y-coordinate of the top-left corner of the component. */
00151     jas_image_coord_t tly_;
00152
00153     /* The horizontal sampling period in units of the reference grid. */
00154     jas_image_coord_t hstep_;
00155

```

```

00179      /* The vertical sampling period in units of the reference grid. */
00180      jas_image_coord_t vstep_;
00181
00182      /* The component width in samples. */
00183      jas_image_coord_t width_;
00184
00185      /* The component height in samples. */
00186      jas_image_coord_t height_;
00187
00188      /* The precision of the sample data (i.e., the number of bits per sample).
00189      If the samples are signed values, this quantity includes the sign bit. */
00190      unsigned prec_;
00191
00192      /* The signedness of the sample data. */
00193      int sgnd_;
00194
00195      /* The stream containing the component data. */
00196      jas_stream_t *stream_;
00197
00198      /* The number of characters per sample in the stream. */
00199      unsigned cps_;
00200
00201      /* The type of component (e.g., opacity, red, green, blue, luma). */
00202      jas_image_cmpttype_t type_;
00203
00204 } jas_image_cmpt_t;
00205
00215 typedef struct {
00216
00217      /* The x-coordinate of the top-left corner of the image bounding box. */
00218      jas_image_coord_t tlx_;
00219
00220      /* The y-coordinate of the top-left corner of the image bounding box. */
00221      jas_image_coord_t tly_;
00222
00223      /* The x-coordinate of the bottom-right corner of the image bounding
00224      box (plus one). */
00225      jas_image_coord_t brx_;
00226
00227      /* The y-coordinate of the bottom-right corner of the image bounding
00228      box (plus one). */
00229      jas_image_coord_t bry_;
00230
00231      /* The number of components. */
00232      unsigned numcmpts_;
00233
00234      /* The maximum number of components that this image can have (i.e., the
00235      allocated size of the components array). */
00236      unsigned maxcmpts_;
00237
00238      /* Per-component information. */
00239      jas_image_cmpt_t **cmpts_;
00240
00241      /* The color space. */
00242      jas_clrspc_t clrspc_;
00243
00244      /* The CM profile. */
00245      jas_cmprof_t *cmprof_;
00246
00247      //bool inmem_;
00248
00249 } jas_image_t;
00250
00258 typedef struct {
00259
00260      /* The x-coordinate of the top-left corner of the component. */
00261      jas_image_coord_t tlx_;
00262
00263      /* The y-coordinate of the top-left corner of the component. */
00264      jas_image_coord_t tly_;
00265
00266      /* The horizontal sampling period in units of the reference grid. */
00267      jas_image_coord_t hstep_;
00268
00269      /* The vertical sampling period in units of the reference grid. */
00270      jas_image_coord_t vstep_;
00271
00272      /* The width of the component in samples. */
00273      jas_image_coord_t width_;
00274
00275      /* The height of the component in samples. */

```

```

00276     jas_image_coord_t height;
00277
00278     /* The precision of the component sample data. */
00279     unsigned prec;
00280
00281     /* The signedness of the component sample data. */
00282     int sgnd;
00283
00284 } jas_image_cmptparm_t;
00285
00286 /*****
00287  * File format related classes.
00288  *****/
00289
00293 #define JAS_IMAGE_MAXFMTS      32
00294
00298 typedef struct {
00299
00301     jas_image_t *(*decode)(jas_stream_t *in, const char *opts);
00302
00304     int (*encode)(jas_image_t *image, jas_stream_t *out, const char *opts);
00305
00307     int (*validate)(jas_stream_t *in);
00308
00309 } jas_image_fmtops_t;
00310
00314 typedef struct {
00315
00317     int id;
00318
00320     char *name;
00321
00322     /* The primary file name extension associated with this format. */
00323     /* This member only exists for backward compatibility. */
00324     char *ext;
00325
00327     char **exts;
00328     size_t max_exts;
00329     size_t num_exts;
00330
00332     int enabled;
00333
00335     char *desc;
00336
00338     jas_image_fmtops_t ops;
00339
00340 } jas_image_fmtinfo_t;
00341
00342 /*****
00343  * Image operations.
00344  *****/
00345
00349 JAS_EXPORT
00350 jas_image_t *jas_image_create(unsigned numcmpts,
00351     const jas_image_cmptparm_t *cmptparms, jas_clrspc_t clrspc);
00352
00356 JAS_EXPORT
00357 jas_image_t *jas_image_create0(void);
00358
00362 JAS_EXPORT
00363 jas_image_t *jas_image_copy(jas_image_t *image);
00364
00368 JAS_EXPORT
00369 void jas_image_destroy(jas_image_t *image);
00370
00374 #define jas_image_width(image) \
00375     ((image)->brx_ - (image)->tlx_)
00376
00380 #define jas_image_height(image) \
00381     ((image)->bry_ - (image)->tly_)
00382
00387 #define jas_image_tlx(image) \
00388     ((image)->tlx_)
00389
00394 #define jas_image_tly(image) \
00395     ((image)->tly_)
00396
00401 #define jas_image_brx(image) \
00402     ((image)->brx_)
00403
00408 #define jas_image_bry(image) \

```



```

00409         ((image)->bry_)
00410
00414 #define jas_image_numcmpts(image) \
00415     ((image)->numcmpts_)
00416
00420 #define jas_image_clrspc(image) \
00421     ((image)->clrspc_)
00422
00426 #define jas_image_setclrspc(image, clrspc) \
00427     ((image)->clrspc_ = (clrspc))
00428
00429 #define jas_image_cmpttype(image, cmptno) \
00430     ((image)->cmpts_[cmptno]->type_)
00431 #define jas_image_setcmpttype(image, cmptno, type) \
00432     ((image)->cmpts_[cmptno]->type_ = (type))
00433
00437 #define jas_image_cmptwidth(image, cmptno) \
00438     ((image)->cmpts_[cmptno]->width_)
00439
00443 #define jas_image_cmptheight(image, cmptno) \
00444     ((image)->cmpts_[cmptno]->height_)
00445
00449 #define jas_image_cmptsgnd(image, cmptno) \
00450     ((image)->cmpts_[cmptno]->sgnd_)
00451
00455 #define jas_image_cmptprec(image, cmptno) \
00456     ((image)->cmpts_[cmptno]->prec_)
00457
00461 #define jas_image_cmptvstep(image, cmptno) \
00462     ((image)->cmpts_[cmptno]->hstep_)
00463
00467 #define jas_image_cmptvstep(image, cmptno) \
00468     ((image)->cmpts_[cmptno]->vstep_)
00469
00473 #define jas_image_cmpttlx(image, cmptno) \
00474     ((image)->cmpts_[cmptno]->tlx_)
00475
00479 #define jas_image_cmpttly(image, cmptno) \
00480     ((image)->cmpts_[cmptno]->tly_)
00481
00486 #define jas_image_cmptbrx(image, cmptno) \
00487     ((image)->cmpts_[cmptno]->tlx_ + (image)->cmpts_[cmptno]->width_ * \
00488     (image)->cmpts_[cmptno]->hstep_)
00489
00494 #define jas_image_cmptbry(image, cmptno) \
00495     ((image)->cmpts_[cmptno]->tly_ + (image)->cmpts_[cmptno]->height_ * \
00496     (image)->cmpts_[cmptno]->vstep_)
00497
00501 JAS_ATTRIBUTE_PURE
00502 JAS_EXPORT
00503 bool jas_image_cmpt_domains_same(const jas_image_t *image);
00504
00509 JAS_ATTRIBUTE_PURE
00510 JAS_EXPORT
00511 uint_fast32_t jas_image_rawsize(const jas_image_t *image);
00512
00516 JAS_EXPORT
00517 jas_image_t *jas_image_decode(jas_stream_t *in, int fmt, const char *optstr);
00518
00522 JAS_EXPORT
00523 int jas_image_encode(jas_image_t *image, jas_stream_t *out, int fmt,
00524     const char *optstr);
00525
00533 JAS_EXPORT
00534 int jas_image_readcmpt(jas_image_t *image, unsigned cmptno,
00535     jas_image_coord_t x, jas_image_coord_t y, jas_image_coord_t width,
00536     jas_image_coord_t height, jas_matrix_t *data);
00537
00541 JAS_EXPORT
00542 int jas_image_writecmpt(jas_image_t *image, unsigned cmptno,
00543     jas_image_coord_t x, jas_image_coord_t y, jas_image_coord_t width,
00544     jas_image_coord_t height, const jas_matrix_t *data);
00545
00549 JAS_EXPORT
00550 void jas_image_delcmpt(jas_image_t *image, unsigned cmptno);
00551
00555 JAS_EXPORT
00556 int jas_image_addcmpt(jas_image_t *image, int cmptno,
00557     const jas_image_cmptparm_t *cmptparm);
00558
00562 JAS_EXPORT

```

```

00563 int jas_image_copycmt(jas_image_t *dstimage, unsigned dstcmtno,
00564     jas_image_t *srcimage, unsigned srccmtno);
00565
00566 JAS_ATTRIBUTE_CONST
00567 static inline bool JAS_IMAGE_CDT_GETSGND(uint_least8_t dtype)
00568 {
00569     return (dtype >> 7) & 1;
00570 }
00571
00572 JAS_ATTRIBUTE_CONST
00573 static inline uint_least8_t JAS_IMAGE_CDT_SETSGND(bool sgnd)
00574 {
00575     return (uint_least8_t)sgnd << 7;
00576 }
00577
00578 JAS_ATTRIBUTE_CONST
00579 static inline uint_least8_t JAS_IMAGE_CDT_GETPREC(uint_least8_t dtype)
00580 {
00581     return dtype & 0x7f;
00582 }
00583
00584 JAS_ATTRIBUTE_CONST
00585 static inline uint_least8_t JAS_IMAGE_CDT_SETPREC(uint_least8_t dtype)
00586 {
00587     return dtype & 0x7f;
00588 }
00589
00590 JAS_ATTRIBUTE_PURE
00591 static inline uint_least8_t jas_image_cmptdtype(const jas_image_t *image,
00592     unsigned cmptno)
00593 {
00594     return JAS_IMAGE_CDT_SETSGND(image->cmpts[cmptno]->sgnd_) |
00595         JAS_IMAGE_CDT_SETPREC(image->cmpts[cmptno]->prec_);
00596 }
00597
00601 JAS_EXPORT
00602 int jas_image_depalettize(jas_image_t *image, unsigned cmptno,
00603     unsigned numlutents, const int_fast32_t *lutents, unsigned dtype,
00604     unsigned newcmtno);
00605
00609 JAS_EXPORT
00610 int jas_image_readcmptsample(jas_image_t *image, unsigned cmptno, unsigned x,
00611     unsigned y);
00612
00616 JAS_EXPORT
00617 void jas_image_writecmptsample(jas_image_t *image, unsigned cmptno,
00618     unsigned x, unsigned y, int_fast32_t v);
00619
00623 JAS_ATTRIBUTE_PURE
00624 JAS_EXPORT
00625 int jas_image_getcmptbytype(const jas_image_t *image, jas_image_cmpttype_t ctype);
00626
00627 /*****
00628  * Image format-related operations.
00629  *****/
00630
00634 JAS_EXPORT
00635 void jas_image_clearfmts(void);
00636
00637 #if defined(JAS_FOR_INTERNAL_USE_ONLY)
00638 void jas_image_clearfmts_internal(jas_image_fmtinfo_t *image_fmtinfos,
00639     size_t *image_numfmts);
00640 #endif
00641
00645 JAS_EXPORT
00646 const jas_image_fmtinfo_t *jas_image_getfmtbyind(int index);
00647
00651 JAS_EXPORT
00652 int jas_image_getnumfmts(void);
00653
00654 #if 0
00655 JAS_EXPORT
00656 int jas_image_delfmtbyid(int id);
00657 #endif
00658
00666 JAS_EXPORT
00667 int jas_image_setfmtenable(int index, int enabled);
00668
00669 #if 0
00670 // TODO: should this be added?
00671 JAS_EXPORT

```

```

00672 int jas_image_getfmtindbyname(const char* name);
00673 #endif
00674
00675 JAS_EXPORT
00676 int jas_image_addfmt(int id, const char *name, const char *ext,
00677     const char *desc, const jas_image_fmtops_t *ops);
00678
00679 #if defined(JAS_FOR_INTERNAL_USE_ONLY)
00680 int jas_image_addfmt_internal(jas_image_fmtinfo_t *image_fmtinfos,
00681     size_t *image_numfmts, int id, const char *name, const char *ext,
00682     const char *desc, const jas_image_fmtops_t *ops);
00683 #endif
00684
00685 JAS_ATTRIBUTE_PURE
00686 JAS_EXPORT
00687 int jas_image_strtofmt(const char *s);
00688
00689 JAS_ATTRIBUTE_CONST
00690 JAS_EXPORT
00691 const char *jas_image_fmtostr(int fmt);
00692
00693 JAS_ATTRIBUTE_CONST
00694 JAS_EXPORT
00695 const jas_image_fmtinfo_t *jas_image_lookupfmtbyid(int id);
00696
00697 JAS_ATTRIBUTE_PURE
00698 JAS_EXPORT
00699 const jas_image_fmtinfo_t *jas_image_lookupfmtbyname(const char *name);
00700
00701 JAS_ATTRIBUTE_PURE
00702 JAS_EXPORT
00703 int jas_image_fmtfromname(const char *filename);
00704
00705 JAS_ATTRIBUTE_PURE
00706 JAS_EXPORT
00707 int jas_image_getfmt(jas_stream_t *in);
00708
00709 #define jas_image_cmprof(image) ((image)->cmprof_)
00710
00711 JAS_ATTRIBUTE_PURE
00712 JAS_EXPORT
00713 int jas_image_ishomosamp(const jas_image_t *image);
00714
00715 JAS_EXPORT
00716 int jas_image_sampcmpt(jas_image_t *image, unsigned cmptno, unsigned newcmptno,
00717     jas_image_coord_t ho, jas_image_coord_t vo, jas_image_coord_t hs,
00718     jas_image_coord_t vs, int sgnd, unsigned prec);
00719
00720 JAS_EXPORT
00721 int jas_image_writecmpt2(jas_image_t *image, unsigned cmptno,
00722     jas_image_coord_t x, jas_image_coord_t y, jas_image_coord_t width,
00723     jas_image_coord_t height, const long *buf);
00724
00725 JAS_EXPORT
00726 int jas_image_readcmpt2(jas_image_t *image, unsigned cmptno,
00727     jas_image_coord_t x, jas_image_coord_t y, jas_image_coord_t width,
00728     jas_image_coord_t height, long *buf);
00729
00730 #define jas_image_setcmprof(image, cmprof) ((image)->cmprof_ = cmprof)
00731
00732 JAS_EXPORT
00733 jas_image_t *jas_image_chclrspc(jas_image_t *image,
00734     const jas_cmprof_t *outprof, jas_cmxform_intent_t intent);
00735
00736 JAS_EXPORT
00737 int jas_image_dump(jas_image_t *image, FILE *out);
00738
00739 /*****
00740  * Image format-dependent operations.
00741  *****/
00742 #if defined(JAS_INCLUDE_JPG_CODECS)
00743 /* Format-dependent operations for JPG support. */
00744 //JAS_EXPORT
00745 jas_image_t *jpg_decode(jas_stream_t *in, const char *optstr);
00746 //JAS_EXPORT
00747 int jpg_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
00748 //JAS_EXPORT
00749 int jpg_validate(jas_stream_t *in);
00750 #endif
00751

```

```
00802 #if defined(JAS_INCLUDE_HEIC_CODEC)
00803 /* Format-dependent operations for HEIC support. */
00804 //JAS_EXPORT
00805 jas_image_t *jas_heic_decode(jas_stream_t *in, const char *optstr);
00806 //JAS_EXPORT
00807 int jas_heic_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
00808 //JAS_EXPORT
00809 int jas_heic_validate(jas_stream_t *in);
00810 #endif
00811
00812 #if defined(JAS_INCLUDE_MIF_CODEC)
00813 /* Format-dependent operations for MIF support. */
00814 //JAS_EXPORT
00815 jas_image_t *mif_decode(jas_stream_t *in, const char *optstr);
00816 //JAS_EXPORT
00817 int mif_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
00818 //JAS_EXPORT
00819 int mif_validate(jas_stream_t *in);
00820 #endif
00821
00822 #if defined(JAS_INCLUDE_PNM_CODEC)
00823 /* Format-dependent operations for PNM support. */
00824 //JAS_EXPORT
00825 jas_image_t *pnm_decode(jas_stream_t *in, const char *optstr);
00826 //JAS_EXPORT
00827 int pnm_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
00828 //JAS_EXPORT
00829 int pnm_validate(jas_stream_t *in);
00830 #endif
00831
00832 #if defined(JAS_INCLUDE_RAS_CODEC)
00833 /* Format-dependent operations for Sun Rasterfile support. */
00834 //JAS_EXPORT
00835 jas_image_t *ras_decode(jas_stream_t *in, const char *optstr);
00836 //JAS_EXPORT
00837 int ras_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
00838 //JAS_EXPORT
00839 int ras_validate(jas_stream_t *in);
00840 #endif
00841
00842 #if defined(JAS_INCLUDE_BMP_CODEC)
00843 /* Format-dependent operations for BMP support. */
00844 //JAS_EXPORT
00845 jas_image_t *bmp_decode(jas_stream_t *in, const char *optstr);
00846 //JAS_EXPORT
00847 int bmp_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
00848 //JAS_EXPORT
00849 int bmp_validate(jas_stream_t *in);
00850 #endif
00851
00852 #if defined(JAS_INCLUDE_JP2_CODEC)
00853 /* Format-dependent operations for JP2 support. */
00854 //JAS_EXPORT
00855 jas_image_t *jp2_decode(jas_stream_t *in, const char *optstr);
00856 //JAS_EXPORT
00857 int jp2_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
00858 //JAS_EXPORT
00859 int jp2_validate(jas_stream_t *in);
00860 #endif
00861
00862 #if defined(JAS_INCLUDE_JPC_CODEC)
00863 /* Format-dependent operations for JPEG-2000 code stream support. */
00864 //JAS_EXPORT
00865 jas_image_t *jpc_decode(jas_stream_t *in, const char *optstr);
00866 //JAS_EXPORT
00867 int jpc_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
00868 //JAS_EXPORT
00869 int jpc_validate(jas_stream_t *in);
00870 #endif
00871
00872 #if defined(JAS_INCLUDE_PGX_CODEC)
00873 /* Format-dependent operations for PGX support. */
00874 //JAS_EXPORT
00875 jas_image_t *pgx_decode(jas_stream_t *in, const char *optstr);
00876 //JAS_EXPORT
00877 int pgx_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
00878 //JAS_EXPORT
00879 int pgx_validate(jas_stream_t *in);
00880 #endif
00881
00885
```

```

00886 #ifdef __cplusplus
00887 }
00888 #endif
00889
00890 #endif

```

16.19 jas_init.h File Reference

JasPer Initialization/Cleanup Code.

```

#include <jasper/jas_config.h>
#include "jasper/jas_malloc.h"
#include "jasper/jas_image.h"
#include "jasper/jas_log.h"
#include <stdarg.h>

```

Typedefs

- typedef void * [jas_context_t](#)
An opaque handle type used to represent a Jasper library context.

Functions

- JAS_EXPORT void [jas_conf_clear](#) (void)
Configure the Jasper library with the default configuration settings.
- JAS_EXPORT void [jas_conf_set_multithread](#) (int multithread)
Set the multithreading flag for the library.
- JAS_EXPORT void [jas_conf_set_allocator](#) ([jas_allocator_t](#) *allocator)
Set the memory allocator to be used by the library.
- JAS_EXPORT void [jas_conf_set_debug_level](#) (int debug_level)
Set the initial debug level for the library.
- JAS_EXPORT void [jas_conf_set_max_mem_usage](#) (size_t max_mem)
Set the maximum amount of memory that can be used by the library (assuming the allocator wrapper is not disabled).
- JAS_EXPORT void [jas_conf_set_dec_default_max_samples](#) (size_t max_samples)
Set the default value for the maximum number of samples that is allowed in an image to be decoded.
- JAS_EXPORT void [jas_conf_set_vlogmsgf](#) ([jas_vlogmsgf_t](#) *func)
Set the function used by the library to output error, warning, and informational messages.
- JAS_EXPORT int [jas_init_library](#) (void)
Initialize the Jasper library with the current configuration settings.
- JAS_EXPORT int [jas_cleanup_library](#) (void)
Perform clean up for the Jasper library.
- JAS_EXPORT int [jas_init_thread](#) (void)
Perform per-thread initialization for the Jasper library.
- JAS_EXPORT int [jas_cleanup_thread](#) (void)
Perform per-thread cleanup for the Jasper library.
- JAS_EXPORT int [jas_init](#) (void)

Configure and initialize the Jasper library using the default configuration settings.

- JAS_EXPORT void [jas_cleanup](#) (void)
Perform any clean up for the Jasper library.
- JAS_EXPORT [jas_context_t](#) [jas_context_create](#) (void)
Create a context.
- JAS_EXPORT void [jas_context_destroy](#) ([jas_context_t](#) context)
Destroy a context.
- JAS_EXPORT [jas_context_t](#) [jas_get_default_context](#) (void)
Get the current context for the calling thread.
- JAS_EXPORT [jas_context_t](#) [jas_get_context](#) (void)
Get the current context for the calling thread.
- JAS_EXPORT void [jas_set_context](#) ([jas_context_t](#) context)
Set the current context for the calling thread.
- JAS_EXPORT void [jas_set_debug_level](#) (int debug_level)
Set the debug level for a particular context.
- static int [jas_get_debug_level](#) (void)
Get the debug level for a particular context.
- JAS_EXPORT void [jas_set_dec_default_max_samples](#) (size_t max_samples)
Set the default maximum number of samples that a decoder is permitted to process.
- static size_t [jas_get_dec_default_max_samples](#) (void)
Get the default maximum number of samples that a decoder is permitted to process.
- JAS_EXPORT void [jas_set_vlogmsgf](#) ([jas_vlogmsgf_t](#) *func)
Set the function to be used for log messages.
- static [jas_vlogmsgf_t](#) * [jas_get_vlogmsgf](#) (void)
Get the function to be used for log messages.

16.19.1 Detailed Description

Jasper Initialization/Cleanup Code.

16.20 [jas_init.h](#)

[Go to the documentation of this file.](#)

```

00001 /*
00002  * Copyright (c) 2001-2002 Michael David Adams.
00003  * All rights reserved.
00004  */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007  *
00008  * Jasper License Version 2.0
00009  *
00010  * Copyright (c) 2001-2006 Michael David Adams
00011  * Copyright (c) 1999-2000 Image Power, Inc.
00012  * Copyright (c) 1999-2000 The University of British Columbia
00013  *
00014  * All rights reserved.
00015  *
00016  * Permission is hereby granted, free of charge, to any person (the
00017  * "User") obtaining a copy of this software and associated documentation
00018  * files (the "Software"), to deal in the Software without restriction,
00019  * including without limitation the rights to use, copy, modify, merge,

```

```

00020  * publish, distribute, and/or sell copies of the Software, and to permit
00021  * persons to whom the Software is furnished to do so, subject to the
00022  * following conditions:
00023  *
00024  * 1. The above copyright notices and this permission notice (which
00025  * includes the disclaimer below) shall be included in all copies or
00026  * substantial portions of the Software.
00027  *
00028  * 2. The name of a copyright holder shall not be used to endorse or
00029  * promote products derived from the Software without specific prior
00030  * written permission.
00031  *
00032  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00054  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058  *
00059  * __END_OF_JASPER_LICENSE__
00060  */
00061
00062
00063
00064
00065
00066
00067 #ifndef JAS_INIT_H
00068 #define JAS_INIT_H
00069
00070 /* The configuration header file should be included first. */
00071 #include <jasper/jas_config.h>
00072
00073 #include "jasper/jas_malloc.h"
00074 #include "jasper/jas_image.h"
00075 #include "jasper/jas_log.h"
00076
00077 #include <stdarg.h>
00078
00079 #ifdef __cplusplus
00080 extern "C" {
00081 #endif
00082
00083
00084
00085
00086
00087 #if defined(JAS_FOR_INTERNAL_USE_ONLY)
00088 /*
00089  * Per-thread library context.
00090  */
00091 typedef struct {
00092     /*
00093      * The level of debugging checks/output enabled by the library.
00094      * A larger value corresponds to a greater level of debugging checks/output.
00095      */
00096     int debug_level;
00097
00098     /*
00099      * The function used to output error/warning/informational messages.
00100      */
00101     int (*vlogmsgf)(jas_logtype_t type, const char *format, va_list ap);
00102
00103     jas_vlogmsgf_t *vlogmsgf;
00104
00105     /*
00106      * The image format information to be used to populate the image format
00107      * table.
00108      */

```

```

00109      */
00110      size_t image_numfmts;
00111      jas_image_fmtinfo_t image_fmtinfos[JAS_IMAGE_MAXFMTS];
00112
00113      /* The maximum number of samples allowable in an image to be decoded. */
00114      size_t dec_default_max_samples;
00115
00116 } jas_ctx_t;
00117 #endif
00118
00123 typedef void *jas_context_t;
00124
00125 /*****
00126  * Library Run-Time Configuration.
00127  *****/
00128
00143 JAS_EXPORT
00144 void jas_conf_clear(void);
00145
00151 JAS_EXPORT
00152 void jas_conf_set_multithread(int multithread);
00153
00163 JAS_EXPORT
00164 void jas_conf_set_allocator(jas_allocator_t *allocator);
00165
00171 JAS_EXPORT
00172 void jas_conf_set_debug_level(int debug_level);
00173
00187 JAS_EXPORT
00188 void jas_conf_set_max_mem_usage(size_t max_mem);
00189
00197 JAS_EXPORT
00198 void jas_conf_set_dec_default_max_samples(size_t max_samples);
00199
00207 JAS_EXPORT
00208 void jas_conf_set_vlogmsgf(jas_vlogmsgf_t *func);
00209
00210 /*****
00211  * Library Initialization and Cleanup.
00212  *****/
00213
00232 JAS_EXPORT
00233 int jas_init_library(void);
00234
00247 JAS_EXPORT
00248 int jas_cleanup_library(void);
00249
00250 /*****
00251  * Thread Initialization and Cleanup.
00252  *****/
00253
00267 JAS_EXPORT
00268 int jas_init_thread(void);
00269
00276 JAS_EXPORT
00277 int jas_cleanup_thread(void);
00278
00279 /*****
00280  * Legacy Initialization and Cleanup Functions.
00281  *****/
00282
00309 JAS_EXPORT
00310 int jas_init(void);
00311
00326 JAS_EXPORT
00327 void jas_cleanup(void);
00328
00329 /*****
00330  * Context Management
00331  *****/
00332
00339 JAS_EXPORT
00340 jas_context_t jas_context_create(void);
00341
00349 JAS_EXPORT
00350 void jas_context_destroy(jas_context_t context);
00351
00358 JAS_EXPORT
00359 jas_context_t jas_get_default_context(void);
00360
00367 JAS_EXPORT

```



```

00368 jas_context_t jas_get_context(void);
00369
00376 JAS_EXPORT
00377 void jas_set_context(jas_context_t context);
00378
00379 /*****
00380 * Getting/Setting Context Properties
00381 *****/
00382
00383 /* This function is only for internal use by the library. */
00384 JAS_EXPORT
00385 int jas_get_debug_level_internal(void);
00386
00387 /* This function is only for internal use by the library. */
00388 JAS_EXPORT
00389 size_t jas_get_dec_default_max_samples_internal(void);
00390
00391 /* This function is only for internal use by the library. */
00392 JAS_EXPORT
00393 jas_vlogmsggf_t *jas_get_vlogmsggf_internal(void);
00394
00395 #if defined(JAS_FOR_INTERNAL_USE_ONLY)
00396 #if defined(JAS_HAVE_THREAD_LOCAL)
00397 extern _Thread_local jas_ctx_t *jas_cur_ctx;
00398 #endif
00399
00400 /* This function is only for internal use by the library. */
00401 jas_ctx_t *jas_get_ctx_internal(void);
00402
00403 /* This function is only for internal use by the library. */
00404 static inline jas_ctx_t *jas_get_ctx(void)
00405 {
00406     #if defined(JAS_HAVE_THREAD_LOCAL)
00407         return jas_cur_ctx ? jas_cur_ctx : jas_get_ctx_internal();
00408     #else
00409         return JAS_CAST(jas_ctx_t *, jas_get_ctx_internal());
00410     #endif
00411 }
00412 #endif
00413
00420 JAS_EXPORT
00421 void jas_set_debug_level(int debug_level);
00422
00429 static inline int jas_get_debug_level(void)
00430 {
00431     #if defined(JAS_FOR_INTERNAL_USE_ONLY)
00432         jas_ctx_t *ctx = jas_get_ctx();
00433         return ctx->debug_level;
00434     #else
00435         return jas_get_debug_level_internal();
00436     #endif
00437 }
00438
00446 JAS_EXPORT
00447 void jas_set_dec_default_max_samples(size_t max_samples);
00448
00456 static inline size_t jas_get_dec_default_max_samples(void)
00457 {
00458     #if defined(JAS_FOR_INTERNAL_USE_ONLY)
00459         jas_ctx_t *ctx = jas_get_ctx();
00460         return ctx->dec_default_max_samples;
00461     #else
00462         return jas_get_dec_default_max_samples_internal();
00463     #endif
00464 }
00465
00472 JAS_EXPORT
00473 void jas_set_vlogmsggf(jas_vlogmsggf_t *func);
00474
00481 static inline
00482 jas_vlogmsggf_t *jas_get_vlogmsggf(void)
00483 {
00484     #if defined(JAS_FOR_INTERNAL_USE_ONLY)
00485         jas_ctx_t *ctx = jas_get_ctx();
00486         return ctx->vlogmsggf;
00487     #else
00488         return jas_get_vlogmsggf_internal();
00489     #endif
00490 }
00491
00495

```

```

00496 #ifdef __cplusplus
00497 }
00498 #endif
00499
00500 #endif

```

16.21 jas_log.h File Reference

JasPer Logging Functionality.

```

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

```

Macros

- #define [JAS_LOGTYPE_CLASS_NULL](#) 0
- #define [JAS_LOGTYPE_CLASS_ERROR](#) 1
- #define [JAS_LOGTYPE_CLASS_WARN](#) 2
- #define [JAS_LOGTYPE_CLASS_INFO](#) 3
- #define [JAS_LOGTYPE_CLASS_DEBUG](#) 4

Typedefs

- typedef int [jas_vlogmsgf_t](#)([jas_logtype_t](#), const char *, va_list)
Type used for formatted message logging function.

Functions

- static [jas_logtype_t](#) [jas_logtype_init](#) (int clas, int priority)
Create an instance of a logtype.
- static int [jas_logtype_getclass](#) ([jas_logtype_t](#) type)
Get the class of a logtype.
- static int [jas_logtype_getpriority](#) ([jas_logtype_t](#) type)
Get the priority of a logtype.
- JAS_EXPORT int [jas_vlogmsgf](#) ([jas_logtype_t](#) type, const char *fmt, va_list ap)
Print formatted log message.
- JAS_EXPORT int [jas_vlogmsgf_stderr](#) ([jas_logtype_t](#) type, const char *fmt, va_list ap)
Output a log message to standard error.
- JAS_EXPORT int [jas_vlogmsgf_discard](#) ([jas_logtype_t](#) type, const char *fmt, va_list ap)
Output a log message to nowhere (i.e., discard the message).

16.21.1 Detailed Description

JasPer Logging Functionality.

16.22 jas_log.h

[Go to the documentation of this file.](#)

```

00001 /*
00002  * Copyright (c) 2001-2002 Michael David Adams.
00003  * All rights reserved.
00004  */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007  *
00008  * JasPer License Version 2.0
00009  *
00010  * Copyright (c) 2001-2006 Michael David Adams
00011  * Copyright (c) 1999-2000 Image Power, Inc.
00012  * Copyright (c) 1999-2000 The University of British Columbia
00013  *
00014  * All rights reserved.
00015  *
00016  * Permission is hereby granted, free of charge, to any person (the
00017  * "User") obtaining a copy of this software and associated documentation
00018  * files (the "Software"), to deal in the Software without restriction,
00019  * including without limitation the rights to use, copy, modify, merge,
00020  * publish, distribute, and/or sell copies of the Software, and to permit
00021  * persons to whom the Software is furnished to do so, subject to the
00022  * following conditions:
00023  *
00024  * 1. The above copyright notices and this permission notice (which
00025  * includes the disclaimer below) shall be included in all copies or
00026  * substantial portions of the Software.
00027  *
00028  * 2. The name of a copyright holder shall not be used to endorse or
00029  * promote products derived from the Software without specific prior
00030  * written permission.
00031  *
00032  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00054  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058  *
00059  * __END_OF_JASPER_LICENSE__
00060  */
00061
00062 #ifndef JAS_LOG_H
00063 #define JAS_LOG_H
00064
00065 /*****\
00066  * Includes.
00067  \*****/
00068
00069 /* The configuration header file should be included first. */
00070 #include <jasper/jas_config.h>
00071
00072 #include <stdio.h>
00073 #include <stdarg.h>
00074

```

```

00080 #ifdef __cplusplus
00081 extern "C" {
00082 #endif
00083
00084
00085 /*****
00086  * Macros and functions.
00087  *****/
00088
00089 #define JAS_LOGTYPE_CLASS_NULL 0
00090 #define JAS_LOGTYPE_CLASS_ERROR 1
00091 #define JAS_LOGTYPE_CLASS_WARN 2
00092 #define JAS_LOGTYPE_CLASS_INFO 3
00093 #define JAS_LOGTYPE_CLASS_DEBUG 4
00094 #define JAS_LOGTYPE_NUM_CLASSES 5
00095
00096 #define JAS_LOGTYPE_MAX_PRIORITY 16384
00097
00098 // NOTE: without the @struct, jas_logtype_t autolinks are not generated
00099 typedef unsigned int jas_logtype_t;
00100
00101 typedef int (jas_vlogmsgf_t)(jas_logtype_t, const char *, va_list);
00102
00103 static inline jas_logtype_t jas_logtype_init(int clas, int priority)
00104 {
00105     assert(clas >= 0 && clas < JAS_LOGTYPE_NUM_CLASSES);
00106     assert(priority >= 0 && priority <= JAS_LOGTYPE_MAX_PRIORITY);
00107     return (clas & 0xf) | (priority << 4);
00108 }
00109
00110 static inline int jas_logtype_getclass(jas_logtype_t type)
00111 {
00112     return type & 0xf;
00113 }
00114
00115 static inline int jas_logtype_getpriority(jas_logtype_t type)
00116 {
00117     return type >> 4;
00118 }
00119
00120 JAS_EXPORT
00121 int jas_vlogmsgf(jas_logtype_t type, const char *fmt, va_list ap);
00122
00123 JAS_EXPORT
00124 int jas_vlogmsgf_stderr(jas_logtype_t type, const char *fmt, va_list ap);
00125
00126 JAS_EXPORT
00127 int jas_vlogmsgf_discard(jas_logtype_t type, const char *fmt, va_list ap);
00128
00129 #ifdef __cplusplus
00130 }
00131 #endif
00132 #endif

```

16.23 jas_malloc.h File Reference

JasPer Memory Allocator.

```

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

```

Classes

- struct [jas_allocator_s](#)

A memory allocator.

- struct `jas_std_allocator_t`

The standard library allocator (i.e., a wrapper for malloc and friends).

Typedefs

- typedef struct `jas_allocator_s` `jas_allocator_t`

A memory allocator.

Functions

- JAS_EXPORT void * `jas_malloc` (size_t size)
Allocate memory.
- JAS_EXPORT void `jas_free` (void *ptr)
Free memory.
- JAS_EXPORT void * `jas_realloc` (void *ptr, size_t size)
Resize a block of allocated memory.
- JAS_EXPORT void * `jas_calloc` (size_t num_elements, size_t element_size)
Allocate a block of memory and initialize the contents to zero.
- JAS_EXPORT void * `jas_alloc2` (size_t num_elements, size_t element_size)
Allocate array (with overflow checking).
- JAS_EXPORT void * `jas_alloc3` (size_t num_arrays, size_t array_size, size_t element_size)
Allocate array of arrays (with overflow checking).
- JAS_EXPORT void * `jas_realloc2` (void *ptr, size_t num_elements, size_t element_size)
Resize a block of allocated memory (with overflow checking).
- JAS_EXPORT void `jas_set_max_mem_usage` (size_t max_mem)
Set the maximum memory usage allowed by the allocator wrapper.
- JAS_EXPORT size_t `jas_get_mem_usage` (void)
Get the current memory usage from the allocator wrapper.
- JAS_EXPORT void `jas_std_allocator_init` (`jas_std_allocator_t` *allocator)
Initialize a memory allocator that uses malloc and related functions for managing memory.
- JAS_EXPORT void `jas_allocator_cleanup` (`jas_allocator_t` *allocator)
Clean up an allocator that is no longer needed.
- JAS_EXPORT size_t `jas_get_total_mem_size` (void)
Get the total amount of memory available on the system.

16.23.1 Detailed Description

JasPer Memory Allocator.

16.24 jas_malloc.h

[Go to the documentation of this file.](#)

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * Jasper License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00064
00065
00066
00067
00068
00069
00070 #ifndef JAS_MALLOC_H
00071 #define JAS_MALLOC_H
00072
00073 /*****\
00074  * Includes.
00075  \*****/
00076
00077 /* The configuration header file should be included first. */
00078 #include <jasper/jas_config.h>
00079

```

```

00080 #include <jasper/jas_types.h>
00081 #include <jasper/jas_thread.h>
00082
00083 #include <stdio.h>
00084
00085 #ifdef __cplusplus
00086 extern "C" {
00087 #endif
00088
00093
00094 /*****
00095  * Types.
00096  *****/
00097
00101 typedef struct jas_allocator_s {
00102
00109     void (*cleanup)(struct jas_allocator_s *allocator);
00110
00115     void *(*alloc)(struct jas_allocator_s *allocator, size_t size);
00116
00121     void (*free)(struct jas_allocator_s *allocator, void *pointer);
00122
00127     void *(*realloc)(struct jas_allocator_s *allocator, void *pointer,
00128                     size_t new_size);
00129
00131     void (*reserved[4])(void);
00132
00133 } jas_allocator_t;
00134
00143 typedef struct {
00144
00145     /* The base class. */
00146     jas_allocator_t base;
00147
00148 } jas_std_allocator_t;
00149
00150 #if defined(JAS_FOR_INTERNAL_USE_ONLY)
00151 /*
00152  The allocator wrapper type.
00153  This type is an allocator that adds memory usage tracking to another
00154  allocator.
00155  The allocator wrapper does not directly perform memory allocation itself.
00156  Instead, it delegate to another allocator.
00157  */
00158 typedef struct {
00159
00160     /* The base class. */
00161     jas_allocator_t base;
00162
00163     /* The delegated-to allocator. */
00164     jas_allocator_t *delegate;
00165
00166     /* The maximum amount of memory that can be used by the allocator. */
00167     size_t max_mem;
00168
00169     /* The current amount of memory in use by the allocator. */
00170     size_t mem;
00171
00172 #if defined(JAS_THREADS)
00173     /* A mutex for synchronized access to the allocator. */
00174     jas_mutex_t mutex;
00175 #endif
00176
00177 } jas_basic_allocator_t;
00178 #endif
00179
00180 /*****
00181  * Data.
00182  *****/
00183
00184 #if defined(JAS_FOR_INTERNAL_USE_ONLY)
00185 extern jas_allocator_t *jas_allocator;
00186 extern jas_std_allocator_t jas_std_allocator;
00187 extern jas_basic_allocator_t jas_basic_allocator;
00188 #endif
00189
00190 /*****
00191  * Functions.
00192  *****/
00193
00203 JAS_EXPORT

```

```

00204 void *jas_malloc(size_t size);
00205
00213 JAS_EXPORT
00214 void jas_free(void *ptr);
00215
00224 JAS_EXPORT
00225 void *jas_realloc(void *ptr, size_t size);
00226
00235 JAS_EXPORT
00236 void *jas_calloc(size_t num_elements, size_t element_size);
00237
00242 JAS_EXPORT
00243 void *jas_alloc2(size_t num_elements, size_t element_size);
00244
00249 JAS_EXPORT
00250 void *jas_alloc3(size_t num_arrays, size_t array_size, size_t element_size);
00251
00256 JAS_EXPORT
00257 void *jas_realloc2(void *ptr, size_t num_elements, size_t element_size);
00258
00280 JAS_EXPORT
00281 void jas_set_max_mem_usage(size_t max_mem);
00282
00295 JAS_EXPORT
00296 size_t jas_get_mem_usage(void);
00297
00311 JAS_EXPORT
00312 void jas_std_allocator_init(jas_std_allocator_t *allocator);
00313
00323 JAS_EXPORT
00324 void jas_allocator_cleanup(jas_allocator_t *allocator);
00325
00326 #if defined(JAS_FOR_INTERNAL_USE_ONLY)
00327
00328 /* This function is for internal library use only. */
00329 void jas_set_allocator(jas_allocator_t* allocator);
00330
00331 /* This function is for internal library use only. */
00332 void jas_basic_allocator_init(jas_basic_allocator_t *allocator,
00333     jas_allocator_t *delegate, size_t max_mem);
00334
00335 #endif
00336
00352 JAS_EXPORT
00353 size_t jas_get_total_mem_size(void);
00354
00358
00359 #ifdef __cplusplus
00360 }
00361 #endif
00362
00363 #endif

```

16.25 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>

```

16.25.1 Detailed Description

Math-Related Code.

16.26 jas_math.h

[Go to the documentation of this file.](#)

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00064
00065 #ifndef JAS_MATH_H
00066 #define JAS_MATH_H
00067
00068
00069 /******\
00070  * Includes
00071  \*****/
00072
00073 /* The configuration header file should be included first. */
00074 #include <jasper/jas_config.h>
00075
00076 #include <jasper/jas_compiler.h>

```

```

00080 #include <jasper/jas_types.h>
00081
00082 #include <assert.h>
00083 #include <string.h>
00084 #include <stdint.h>
00085 #include <limits.h>
00086
00087 #ifdef __cplusplus
00088 extern "C" {
00089 #endif
00090
00091 /*****
00092  * Macros
00093  *****/
00094
00095 #define JAS_KIBI          JAS_CAST(size_t, 1024)
00096 #define JAS_MEBI          (JAS_KIBI * JAS_KIBI)
00097
00098 /* Compute the absolute value. */
00099 #define JAS_ABS(x) \
00100     ((x) >= 0) ? (x) : (-(x))
00101
00102 /* Compute the minimum of two values. */
00103 #define JAS_MIN(x, y) \
00104     ((x) < (y)) ? (x) : (y)
00105
00106 /* Compute the maximum of two values. */
00107 #define JAS_MAX(x, y) \
00108     ((x) > (y)) ? (x) : (y)
00109
00110 /* Compute the remainder from division (where division is defined such
00111    that the remainder is always nonnegative). */
00112 #define JAS_MOD(x, y) \
00113     (((x) < 0) ? (((-x) % (y)) ? ((y) - ((-x) % (y))) : (0)) : ((x) % (y)))
00114
00115 /* Compute the integer with the specified number of least significant bits
00116    set to one. */
00117 #define JAS_ONES(n) \
00118     ((1 << (n)) - 1)
00119 #if 0
00120 #define JAS_ONES_X(type, n) \
00121     ((JAS_CAST(type, 1) << (n)) - 1)
00122 #endif
00123 #define JAS_POW2_X(type, n) \
00124     (JAS_CAST(type, 1) << (n))
00125
00126 /*****
00127  *
00128  *****/
00129
00130 #if defined(__clang__) || (defined(__GNUC__) && __GNUC__ > 6)
00131 /* suppress clang warning "shifting a negative signed value is
00132    undefined" in the assertions below */
00133 #pragma GCC diagnostic push
00134 #pragma GCC diagnostic ignored "-Wshift-negative-value"
00135 #endif
00136
00137 JAS_ATTRIBUTE_CONST
00138 JAS_ATTRIBUTE_DISABLE_UBSAN
00139 inline static int jas_int_asr(int x, unsigned n)
00140 {
00141     // Ensure that the shift of a negative value appears to behave as a
00142     // signed arithmetic shift.
00143     assert(((-1) >> 1) == -1);
00144     // The behavior is undefined when x is negative.
00145     // We tacitly assume the behavior is equivalent to a signed
00146     // arithmetic right shift.
00147     return x >> n;
00148 }
00149
00150 JAS_ATTRIBUTE_CONST
00151 JAS_ATTRIBUTE_DISABLE_UBSAN
00152 inline static int jas_int_asl(int x, unsigned n)
00153 {
00154     // Ensure that the shift of a negative value appears to behave as a
00155     // signed arithmetic shift.
00156     assert(((-1) << 1) == -2);
00157     // The behavior is undefined when x is negative.
00158     // We tacitly assume the behavior is equivalent to a signed
00159     // arithmetic left shift.
00160     return x << n;

```

```

00161 }
00162
00163 JAS_ATTRIBUTE_CONST
00164 JAS_ATTRIBUTE_DISABLE_UBSAN
00165 inline static int_least32_t jas_least32_asr(int_least32_t x, unsigned n)
00166 {
00167     // Ensure that the shift of a negative value appears to behave as a
00168     // signed arithmetic shift.
00169     assert(((JAS_CAST(int_least32_t, -1)) >> 1) == JAS_CAST(int_least32_t, -1));
00170     // The behavior is undefined when x is negative.
00171     // We tacitly assume the behavior is equivalent to a signed
00172     // arithmetic right shift.
00173     return x >> n;
00174 }
00175
00176 JAS_ATTRIBUTE_CONST
00177 JAS_ATTRIBUTE_DISABLE_UBSAN
00178 inline static int_least32_t jas_least32_asl(int_least32_t x, unsigned n)
00179 {
00180     // Ensure that the shift of a negative value appears to behave as a
00181     // signed arithmetic shift.
00182     assert(((JAS_CAST(int_least32_t, -1)) << 1) == JAS_CAST(int_least32_t, -2));
00183     // The behavior is undefined when x is negative.
00184     // We tacitly assume the behavior is equivalent to a signed
00185     // arithmetic left shift.
00186     return x << n;
00187 }
00188
00189 JAS_ATTRIBUTE_CONST
00190 JAS_ATTRIBUTE_DISABLE_UBSAN
00191 inline static int_fast32_t jas_fast32_asr(int_fast32_t x, unsigned n)
00192 {
00193     // Ensure that the shift of a negative value appears to behave as a
00194     // signed arithmetic shift.
00195     assert(((JAS_CAST(int_fast32_t, -1)) >> 1) == JAS_CAST(int_fast32_t, -1));
00196     // The behavior is undefined when x is negative.
00197     // We tacitly assume the behavior is equivalent to a signed
00198     // arithmetic right shift.
00199     return x >> n;
00200 }
00201
00202 JAS_ATTRIBUTE_CONST
00203 JAS_ATTRIBUTE_DISABLE_UBSAN
00204 inline static int_fast32_t jas_fast32_asl(int_fast32_t x, unsigned n)
00205 {
00206     // Ensure that the shift of a negative value appears to behave as a
00207     // signed arithmetic shift.
00208     assert(((JAS_CAST(int_fast32_t, -1)) << 1) == JAS_CAST(int_fast32_t, -2));
00209     // The behavior is undefined when x is negative.
00210     // We tacitly assume the behavior is equivalent to a signed
00211     // arithmetic left shift.
00212     return x << n;
00213 }
00214
00215 #if defined(__clang__) || (defined(__GNUC__) && __GNUC__ > 6)
00216 #pragma GCC diagnostic pop
00217 #endif
00218
00219 /*****
00220 * Safe integer arithmetic (i.e., with overflow checking).
00221 *****/
00222
00223 /* Compute the product of two size_t integers with overflow checking. */
00224 inline static bool jas_safe_size_mul(size_t x, size_t y, size_t *result)
00225 {
00226     #if jas_has_builtin(__builtin_mul_overflow) || (defined(__GNUC__) && __GNUC__ > 5)
00227         size_t result_buffer;
00228         bool valid = !__builtin_mul_overflow(x, y, &result_buffer);
00229         if (valid && result) {
00230             *result = result_buffer;
00231         }
00232         return valid;
00233     #else
00234         /* Check if overflow would occur */
00235         if (x && y > SIZE_MAX / x) {
00236             /* Overflow would occur. */
00237             return false;
00238         }
00239         if (result) {
00240             *result = x * y;
00241         }
00242     }

```

```

00242         return true;
00243 #endif
00244 }
00245
00246 /* Compute the product of three size_t integers with overflow checking. */
00247 inline static bool jas_safe_size_mul3(size_t a, size_t b, size_t c,
00248     size_t *result)
00249 {
00250     size_t tmp;
00251     if (!jas_safe_size_mul(a, b, &tmp) ||
00252         !jas_safe_size_mul(tmp, c, &tmp)) {
00253         return false;
00254     }
00255     if (result) {
00256         *result = tmp;
00257     }
00258     return true;
00259 }
00260
00261 /* Compute the sum of two size_t integers with overflow checking. */
00262 inline static bool jas_safe_size_add(size_t x, size_t y, size_t *result)
00263 {
00264     #if jas_has_builtin(__builtin_add_overflow) || (defined(__GNUC__) && __GNUC__ > 5)
00265         size_t result_buffer;
00266         bool valid = !__builtin_add_overflow(x, y, &result_buffer);
00267         if (valid && result) {
00268             *result = result_buffer;
00269         }
00270         return valid;
00271     #else
00272         if (y > SIZE_MAX - x) {
00273             return false;
00274         }
00275         if (result) {
00276             *result = x + y;
00277         }
00278         return true;
00279     #endif
00280 }
00281
00282 /* Compute the difference of two size_t integers with overflow checking. */
00283 inline static bool jas_safe_size_sub(size_t x, size_t y, size_t *result)
00284 {
00285     #if jas_has_builtin(__builtin_sub_overflow) || (defined(__GNUC__) && __GNUC__ > 5)
00286         size_t result_buffer;
00287         bool valid = !__builtin_sub_overflow(x, y, &result_buffer);
00288         if (valid && result) {
00289             *result = result_buffer;
00290         }
00291         return valid;
00292     #else
00293         if (y > x) {
00294             return false;
00295         }
00296         if (result) {
00297             *result = x - y;
00298         }
00299         return true;
00300     #endif
00301 }
00302
00303 /* Compute the product of two int_fast32_t integers with overflow checking. */
00304 inline static bool jas_safe_intfast32_mul(int_fast32_t x, int_fast32_t y,
00305     int_fast32_t *result)
00306 {
00307     #if jas_has_builtin(__builtin_mul_overflow) || (defined(__GNUC__) && __GNUC__ > 5)
00308         int_fast32_t result_buffer;
00309         bool valid = !__builtin_mul_overflow(x, y, &result_buffer);
00310         if (valid && result) {
00311             *result = result_buffer;
00312         }
00313         return valid;
00314     #else
00315         if (x > 0) {
00316             /* x is positive */
00317             if (y > 0) {
00318                 /* x and y are positive */
00319                 if (x > INT_FAST32_MAX / y) {
00320                     return false;
00321                 }
00322             } else {

```

```

00323             /* x positive, y nonpositive */
00324             if (y < INT_FAST32_MIN / x) {
00325                 return false;
00326             }
00327         }
00328     } else {
00329         /* x is nonpositive */
00330         if (y > 0) {
00331             /* x is nonpositive, y is positive */
00332             if (x < INT_FAST32_MIN / y) {
00333                 return false;
00334             }
00335         } else { /* x and y are nonpositive */
00336             if (x != 0 && y < INT_FAST32_MAX / x) {
00337                 return false;
00338             }
00339         }
00340     }
00341 }
00342 if (result) {
00343     *result = x * y;
00344 }
00345 return true;
00346 #endif
00347 }
00348
00349 /* Compute the product of three int_fast32_t integers with overflow checking. */
00350 inline static bool jas_safe_intfast32_mul(int_fast32_t a, int_fast32_t b,
00351     int_fast32_t c, int_fast32_t *result)
00352 {
00353     int_fast32_t tmp;
00354     if (!jas_safe_intfast32_mul(a, b, &tmp) ||
00355         !jas_safe_intfast32_mul(tmp, c, &tmp)) {
00356         return false;
00357     }
00358     if (result) {
00359         *result = tmp;
00360     }
00361     return true;
00362 }
00363
00364 /* Compute the sum of two int_fast32_t integers with overflow checking. */
00365 inline static bool jas_safe_intfast32_add(int_fast32_t x, int_fast32_t y,
00366     int_fast32_t *result)
00367 {
00368     #if jas_has_builtin(__builtin_add_overflow) || (defined(__GNUC__) && __GNUC__ > 5)
00369         int_fast32_t result_buffer;
00370         bool valid = !__builtin_add_overflow(x, y, &result_buffer);
00371         if (valid && result) {
00372             *result = result_buffer;
00373         }
00374         return valid;
00375     #else
00376         if ((y > 0 && x > INT_FAST32_MAX - y) ||
00377             (y < 0 && x < INT_FAST32_MIN - y)) {
00378             return false;
00379         }
00380         if (result) {
00381             *result = x + y;
00382         }
00383         return true;
00384     #endif
00385 }
00386
00387 #if 0
00388 /*
00389 This function is potentially useful but not currently used.
00390 So, it is commented out.
00391 */
00392 inline static bool jas_safe_uint_mul(unsigned x, unsigned y, unsigned *result)
00393 {
00394     /* Check if overflow would occur */
00395     if (x && y > UINT_MAX / x) {
00396         /* Overflow would occur. */
00397         return false;
00398     }
00399     if (result) {
00400         *result = x * y;
00401     }
00402     return true;
00403 }

```

```

00404 #endif
00405
00406 /*****
00407  * Safe 32-bit unsigned integer arithmetic (i.e., with overflow checking).
00408  *****/
00409
00410 #define JAS_SAFEUI32_MAX (0xffffffffUL)
00411
00412 typedef struct {
00413     bool valid;
00414     uint_least32_t value;
00415 } jas_safeui32_t;
00416
00417 JAS_ATTRIBUTE_CONST
00418 static inline jas_safeui32_t jas_safeui32_from_ulong(unsigned long x)
00419 {
00420     jas_safeui32_t result;
00421     if (x <= JAS_SAFEUI32_MAX) {
00422         result.valid = 1;
00423         result.value = JAS_CAST(uint_least32_t, x);
00424     } else {
00425         result.valid = 0;
00426         result.value = 0;
00427     }
00428     return result;
00429 }
00430
00431 JAS_ATTRIBUTE_PURE
00432 static inline bool jas_safeui32_to_intfast32(jas_safeui32_t x,
00433     int_fast32_t* y)
00434 {
00435     const long I32_MAX = 0x7fffffffL;
00436     if (x.value <= I32_MAX) {
00437         *y = x.value;
00438         return true;
00439     } else {
00440         return false;
00441     }
00442 }
00443
00444 JAS_ATTRIBUTE_CONST
00445 static inline jas_safeui32_t jas_safeui32_add(jas_safeui32_t x,
00446     jas_safeui32_t y)
00447 {
00448     jas_safeui32_t result;
00449     if (x.valid && y.valid && y.value <= UINT_LEAST32_MAX - x.value) {
00450         result.valid = true;
00451         result.value = x.value + y.value;
00452     } else {
00453         result.valid = false;
00454         result.value = 0;
00455     }
00456     return result;
00457 }
00458
00459 JAS_ATTRIBUTE_CONST
00460 static inline
00461 jas_safeui32_t jas_safeui32_sub(jas_safeui32_t x, jas_safeui32_t y)
00462 {
00463     jas_safeui32_t result;
00464     if (x.valid && y.valid && y.value <= x.value) {
00465         result.valid = true;
00466         result.value = x.value - y.value;
00467     } else {
00468         result.valid = false;
00469         result.value = 0;
00470     }
00471     return result;
00472 }
00473
00474 JAS_ATTRIBUTE_CONST
00475 static inline jas_safeui32_t jas_safeui32_mul(jas_safeui32_t x,
00476     jas_safeui32_t y)
00477 {
00478     jas_safeui32_t result;
00479     if (!x.valid || !y.valid || (x.value && y.value > UINT_LEAST32_MAX /
00480         x.value)) {
00481         result.valid = false;
00482         result.value = 0;
00483     } else {
00484         result.valid = true;

```

```

00485         result.value = x.value * y.value;
00486     }
00487     return result;
00488 }
00489
00490 /*****
00491 * Safe 64-bit signed integer arithmetic (i.e., with overflow checking).
00492 *****/
00493
00494 typedef struct {
00495     bool valid;
00496     int_least64_t value;
00497 } jas_safei64_t;
00498
00499 JAS_ATTRIBUTE_CONST
00500 static inline
00501 jas_safei64_t jas_safei64_from_intmax(intmax_t x)
00502 {
00503     jas_safei64_t result;
00504     if (x >= INT_LEAST64_MIN && x <= INT_LEAST64_MAX) {
00505         result.valid = true;
00506         result.value = JAS_CAST(int_least64_t, x);
00507     } else {
00508         result.valid = false;
00509         result.value = 0;
00510     }
00511     return result;
00512 }
00513
00514 JAS_ATTRIBUTE_CONST
00515 static inline
00516 jas_safei64_t jas_safei64_add(jas_safei64_t x, jas_safei64_t y)
00517 {
00518     jas_safei64_t result;
00519     if ((y.value > 0 && (x.value > (INT_LEAST64_MAX - y.value))) ||
00520         (y.value < 0 && (x.value < (INT_LEAST64_MIN - y.value)))) {
00521         result.valid = false;
00522         result.value = 0;
00523     } else {
00524         result.valid = true;
00525         result.value = x.value + y.value;
00526     }
00527     return result;
00528 }
00529
00530 JAS_ATTRIBUTE_CONST
00531 static inline
00532 jas_safei64_t jas_safei64_sub(jas_safei64_t x, jas_safei64_t y)
00533 {
00534     jas_safei64_t result;
00535     if ((y.value > 0 && x.value < INT_LEAST64_MIN + y.value) ||
00536         (y.value < 0 && x.value > INT_LEAST64_MAX + y.value)) {
00537         result.valid = false;
00538         result.value = 0;
00539     } else {
00540         result.valid = true;
00541         result.value = x.value - y.value;
00542     }
00543     return result;
00544 }
00545
00546 JAS_ATTRIBUTE_CONST
00547 static inline
00548 jas_safei64_t jas_safei64_mul(jas_safei64_t x, jas_safei64_t y)
00549 {
00550     jas_safei64_t result;
00551     if (x.value > 0) { /* x.value is positive */
00552         if (y.value > 0) { /* x.value and y.value are positive */
00553             if (x.value > (INT_LEAST64_MAX / y.value)) {
00554                 goto error;
00555             }
00556         } else { /* x.value positive, y.value nonpositive */
00557             if (y.value < (INT_LEAST64_MIN / x.value)) {
00558                 goto error;
00559             }
00560         } /* x.value positive, y.value nonpositive */
00561     } else { /* x.value is nonpositive */
00562         if (y.value > 0) { /* x.value is nonpositive, y.value is positive */
00563             if (x.value < (INT_LEAST64_MIN / y.value)) {
00564                 goto error;
00565             }
00566         }
00567     }

```

```

00566         } else { /* x.value and y.value are nonpositive */
00567             if ( (x.value != 0) && (y.value < (INT_LEAST64_MAX / x.value)) ) {
00568                 goto error;
00569             }
00570         } /* End if x.value and y.value are nonpositive */
00571     } /* End if x.value is nonpositive */
00572     result.valid = true;
00573     result.value = x.value * y.value;
00574     return result;
00575 error:
00576     result.valid = false;
00577     result.value = 0;
00578     return result;
00579 }
00580
00581 #if 0
00582 JAS_ATTRIBUTE_CONST
00583 static inline
00584 jas_safei64_t jas_safei64_div(jas_safei64_t x, jas_safei64_t y)
00585 {
00586     // TODO/FIXME: Not yet implemented.
00587     jas_safei64_t result;
00588     result.valid = false;
00589     result.value = 0;
00590     return result;
00591 }
00592 #endif
00593
00594 JAS_ATTRIBUTE_CONST
00595 static inline
00596 jas_i32_t jas_safei64_to_i32(jas_safei64_t x, jas_i32_t invalid_value)
00597 {
00598     jas_i32_t result;
00599     if (x.valid && x.value >= JAS_I32_MIN && x.value <= JAS_I32_MAX) {
00600         result = JAS_CAST(jas_i32_t, x.value);
00601     } else {
00602         result = invalid_value;
00603     }
00604     return result;
00605 }
00606
00607 /*****
00608 * Safe 64-bit unsigned integer arithmetic (i.e., with overflow checking).
00609 *****/
00610
00611 typedef struct {
00612     bool valid;
00613     uint_least64_t value;
00614 } jas_safeui64_t;
00615
00616 JAS_ATTRIBUTE_CONST
00617 static inline
00618 jas_safeui64_t jas_safeui64_from_intmax(intmax_t x)
00619 {
00620     jas_safeui64_t result;
00621     if (x >= 0 && x <= UINT_LEAST64_MAX) {
00622         result.valid = true;
00623         result.value = JAS_CAST(uint_least64_t, x);
00624     } else {
00625         result.valid = false;
00626         result.value = 0;
00627     }
00628     return result;
00629 }
00630
00631 JAS_ATTRIBUTE_CONST
00632 static inline
00633 jas_safeui64_t jas_safeui64_add(jas_safeui64_t x, jas_safeui64_t y)
00634 {
00635     jas_safeui64_t result;
00636     if (x.valid && y.valid && y.value <= UINT_LEAST64_MAX - x.value) {
00637         result.valid = true;
00638         result.value = x.value + y.value;
00639     } else {
00640         result.valid = false;
00641         result.value = 0;
00642     }
00643     return result;
00644 }
00645
00646 JAS_ATTRIBUTE_CONST

```



```

00647 static inline
00648 jas_safeui64_t jas_safeui64_sub(jas_safeui64_t x, jas_safeui64_t y)
00649 {
00650     jas_safeui64_t result;
00651     if (x.valid && y.valid && y.value <= x.value) {
00652         result.valid = true;
00653         result.value = x.value - y.value;
00654     } else {
00655         result.valid = false;
00656         result.value = 0;
00657     }
00658     return result;
00659 }
00660
00661 JAS_ATTRIBUTE_CONST
00662 static inline
00663 jas_safeui64_t jas_safeui64_mul(jas_safeui64_t x, jas_safeui64_t y)
00664 {
00665     jas_safeui64_t result;
00666     if (!x.valid || !y.valid || (x.value && y.value > UINT_LEAST64_MAX /
00667         x.value)) {
00668         result.valid = false;
00669         result.value = 0;
00670     } else {
00671         result.valid = true;
00672         result.value = x.value * y.value;
00673     }
00674     return result;
00675 }
00676
00677 JAS_ATTRIBUTE_CONST
00678 static inline
00679 jas_safeui64_t jas_safeui64_div(jas_safeui64_t x, jas_safeui64_t y)
00680 {
00681     jas_safeui64_t result;
00682     if (x.valid && y.valid && y.value) {
00683         result.valid = true;
00684         result.value = x.value / y.value;
00685     } else {
00686         result.valid = false;
00687         result.value = 0;
00688     }
00689     return result;
00690 }
00691
00692 JAS_ATTRIBUTE_CONST
00693 static inline
00694 jas_safeui64_t jas_safeui64_pow2_intmax(intmax_t x)
00695 {
00696     jas_safeui64_t result;
00697     if (x >= 0 && x < 64) {
00698         result.valid = true;
00699         result.value = JAS_CAST(uint_least64_t, 1) << x;
00700     } else {
00701         result.valid = false;
00702         result.value = 0;
00703     }
00704     return result;
00705 }
00706
00707 JAS_ATTRIBUTE_CONST
00708 static inline
00709 int jas_safeui64_to_int(jas_safeui64_t x, int invalid_value)
00710 {
00711     int result;
00712     if (x.valid && x.value <= INT_MAX) {
00713         result = JAS_CAST(int, x.value);
00714     } else {
00715         result = invalid_value;
00716     }
00717     return result;
00718 }
00719
00720 JAS_ATTRIBUTE_CONST
00721 static inline
00722 jas_ui32_t jas_safeui64_to_ui32(jas_safeui64_t x, jas_ui32_t invalid_value)
00723 {
00724     jas_ui32_t result;
00725     if (x.valid && x.value <= JAS_UI32_MAX) {
00726         result = JAS_CAST(jas_ui32_t, x.value);
00727     } else {

```

```

00728             result = invalid_value;
00729         }
00730         return result;
00731     }
00732
00733     JAS_ATTRIBUTE_CONST
00734     static inline
00735     jas_i32_t jas_safeui64_to_i32(jas_safeui64_t x, jas_i32_t invalid_value)
00736     {
00737         jas_i32_t result;
00738         if (x.valid && x.value >= JAS_I32_MIN && x.value <= JAS_I32_MAX) {
00739             result = JAS_CAST(jas_i32_t, x.value);
00740         } else {
00741             result = invalid_value;
00742         }
00743         return result;
00744     }
00745
00746     /*****
00747     \*****/
00748
00749     #ifdef __cplusplus
00750     }
00751     #endif
00752
00753     #endif

```

16.27 jas_seq.h File Reference

Sequence/Matrix Library.

```

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

```

Classes

- struct [jas_matrix_t](#)
Matrix type.

Functions

- static JAS_ATTRIBUTE_PURE jas_matind_t [jas_matrix_numrows](#) (const [jas_matrix_t](#) *matrix)
Get the number of rows in a matrix.
- static JAS_ATTRIBUTE_PURE jas_matind_t [jas_matrix_numcols](#) (const [jas_matrix_t](#) *matrix)
Get the number of columns in a matrix.
- static JAS_ATTRIBUTE_PURE jas_matind_t [jas_matrix_size](#) (const [jas_matrix_t](#) *matrix)
Get the number of elements in a matrix.
- static JAS_ATTRIBUTE_PURE bool [jas_matrix_empty](#) (const [jas_matrix_t](#) *matrix)
Test if a matrix is empty (i.e., contains no elements).
- static JAS_ATTRIBUTE_PURE jas_seqent_t [jas_matrix_get](#) (const [jas_matrix_t](#) *matrix, jas_matind_t i, jas_matind_t j)
Get a matrix element.

- static void [jas_matrix_set](#) ([jas_matrix_t](#) *matrix, [jas_matind_t](#) i, [jas_matind_t](#) j, [jas_seqent_t](#) v)
Set a matrix element.
- static [JAS_ATTRIBUTE_PURE](#) [jas_seqent_t](#) [jas_matrix_getv](#) (const [jas_matrix_t](#) *matrix, [jas_matind_t](#) i)
Get an element from a matrix that is known to be a row or column vector.
- static void [jas_matrix_setv](#) ([jas_matrix_t](#) *matrix, [jas_matind_t](#) i, [jas_seqent_t](#) v)
Set an element in a matrix that is known to be a row or column vector.
- static [JAS_ATTRIBUTE_PURE](#) [jas_seqent_t](#) * [jas_matrix_getref](#) (const [jas_matrix_t](#) *matrix, [jas_matind_t](#) i, [jas_matind_t](#) j)
Get the address of an element in a matrix.
- static [JAS_ATTRIBUTE_PURE](#) [jas_seqent_t](#) * [jas_matrix_getvref](#) (const [jas_matrix_t](#) *matrix, [jas_matind_t](#) i)
Get a reference to a particular row of a 2-D sequence.
- [JAS_EXPORT](#) [jas_matrix_t](#) * [jas_matrix_create](#) ([jas_matind_t](#) numRows, [jas_matind_t](#) numcols)
Create a matrix with the specified dimensions.
- [JAS_EXPORT](#) void [jas_matrix_destroy](#) ([jas_matrix_t](#) *matrix)
Destroy a matrix.
- [JAS_EXPORT](#) int [jas_matrix_resize](#) ([jas_matrix_t](#) *matrix, [jas_matind_t](#) numRows, [jas_matind_t](#) numcols)
Resize a matrix. The previous contents of the matrix are lost.
- [JAS_EXPORT](#) int [jas_matrix_output](#) ([jas_matrix_t](#) *matrix, FILE *out)
Write a matrix to a C standard library stream.
- [JAS_EXPORT](#) int [jas_matrix_bindsub](#) ([jas_matrix_t](#) *mat0, [jas_matrix_t](#) *mat1, [jas_matind_t](#) r0, [jas_matind_t](#) c0, [jas_matind_t](#) r1, [jas_matind_t](#) c1)
Create a matrix that references part of another matrix.
- static int [jas_matrix_bindrow](#) ([jas_matrix_t](#) *mat0, [jas_matrix_t](#) *mat1, [jas_matind_t](#) r)
Create a matrix that is a reference to a row of another matrix.
- static int [jas_matrix_bindcol](#) ([jas_matrix_t](#) *mat0, [jas_matrix_t](#) *mat1, [jas_matind_t](#) c)
Create a matrix that is a reference to a column of another matrix.
- [JAS_EXPORT](#) void [jas_matrix_clip](#) ([jas_matrix_t](#) *matrix, [jas_seqent_t](#) minval, [jas_seqent_t](#) maxval)
Clip the values of matrix elements to the specified range.
- [JAS_EXPORT](#) void [jas_matrix_asl](#) ([jas_matrix_t](#) *matrix, unsigned n)
Arithmetic shift left of all elements in a matrix.
- [JAS_EXPORT](#) void [jas_matrix_asr](#) ([jas_matrix_t](#) *matrix, unsigned n)
Arithmetic shift right of all elements in a matrix.
- [JAS_EXPORT](#) void [jas_matrix_divpow2](#) ([jas_matrix_t](#) *matrix, unsigned n)
Almost-but-not-quite arithmetic shift right of all elements in a matrix.
- [JAS_EXPORT](#) void [jas_matrix_setall](#) ([jas_matrix_t](#) *matrix, [jas_seqent_t](#) val)
Set all elements of a matrix to the specified value.
- static [JAS_ATTRIBUTE_PURE](#) [size_t](#) [jas_matrix_rowstep](#) (const [jas_matrix_t](#) *matrix)
The spacing between rows of a matrix.
- static [JAS_ATTRIBUTE_PURE](#) [size_t](#) [jas_matrix_step](#) (const [jas_matrix_t](#) *matrix)
The spacing between columns of a matrix.
- [JAS_EXPORT](#) int [jas_matrix_cmp](#) ([jas_matrix_t](#) *mat0, [jas_matrix_t](#) *mat1)
Compare two matrices for equality.
- [JAS_EXPORT](#) [jas_matrix_t](#) * [jas_matrix_copy](#) ([jas_matrix_t](#) *x)
Copy a matrix.
- [JAS_EXPORT](#) [jas_matrix_t](#) * [jas_matrix_input](#) (FILE *)
Read a matrix from a C standard library stream.
- [JAS_EXPORT](#) [jas_seq2d_t](#) * [jas_seq2d_copy](#) ([jas_seq2d_t](#) *x)

Copy a 2-D sequence.

- JAS_EXPORT [jas_matrix_t](#) * [jas_seq2d_create](#) ([jas_matind_t](#) xstart, [jas_matind_t](#) ystart, [jas_matind_t](#) xend, [jas_matind_t](#) yend)

Create a 2-D sequence.

- static void [jas_seq2d_destroy](#) ([jas_seq2d_t](#) *s)

Destroy a 2-D sequence.

- static JAS_ATTRIBUTE_PURE [jas_matind_t](#) [jas_seq2d_xstart](#) (const [jas_seq2d_t](#) *s)

Get the starting x-coordinate of the sequence.

- static JAS_ATTRIBUTE_PURE [jas_matind_t](#) [jas_seq2d_ystart](#) (const [jas_seq2d_t](#) *s)

Get the starting y-coordinate of the sequence.

- static JAS_ATTRIBUTE_PURE [jas_matind_t](#) [jas_seq2d_xend](#) (const [jas_seq2d_t](#) *s)

Get the ending x-coordinate of the sequence.

- static JAS_ATTRIBUTE_PURE [jas_matind_t](#) [jas_seq2d_yend](#) (const [jas_seq2d_t](#) *s)

Get the ending y-coordinate of the sequence.

- static JAS_ATTRIBUTE_PURE [jas_seqent_t](#) * [jas_seq2d_getref](#) (const [jas_seq2d_t](#) *s, [jas_matind_t](#) x, [jas_matind_t](#) y)

Get a pointer (i.e., reference) to an element of a 2-D sequence.

- static JAS_ATTRIBUTE_PURE [jas_seqent_t](#) [jas_seq2d_get](#) (const [jas_seq2d_t](#) *s, [jas_matind_t](#) x, [jas_matind_t](#) y)

Get an element of a 2-D sequence.

- static JAS_ATTRIBUTE_PURE [size_t](#) [jas_seq2d_rowstep](#) (const [jas_seq2d_t](#) *s)

Get the stride between successive rows in the sequence.

- static JAS_ATTRIBUTE_PURE unsigned [jas_seq2d_width](#) (const [jas_seq2d_t](#) *s)

Get the number of columns in the sequence.

- static JAS_ATTRIBUTE_PURE unsigned [jas_seq2d_height](#) (const [jas_seq2d_t](#) *s)

Get the number of rows in the sequence.

- static void [jas_seq2d_setshift](#) ([jas_seq2d_t](#) *s, [jas_matind_t](#) x, [jas_matind_t](#) y)

Set the shift (i.e., starting x- and y-coordinates) of the sequence.

- static JAS_ATTRIBUTE_PURE [jas_matind_t](#) [jas_seq2d_size](#) (const [jas_seq2d_t](#) *s)

Get the number of elements in the sequence.

- static JAS_ATTRIBUTE_PURE bool [jas_seq2d_empty](#) (const [jas_seq2d_t](#) *s)

Test if the sequence is empty (i.e., contains no elements).

- JAS_EXPORT int [jas_seq2d_bindsub](#) ([jas_matrix_t](#) *s, [jas_matrix_t](#) *s1, [jas_matind_t](#) xstart, [jas_matind_t](#) ystart, [jas_matind_t](#) xend, [jas_matind_t](#) yend)

Initialize a sequence to reference a subsequence of another sequence.

- static [jas_seq_t](#) * [jas_seq_create](#) ([jas_matind_t](#) start, [jas_matind_t](#) end)

Create a 1-D sequence.

- static void [jas_seq_destroy](#) ([jas_seq_t](#) *seq)

Destroy a 1-D sequence.

- static void [jas_seq_set](#) ([jas_seq_t](#) *seq, [jas_matind_t](#) i, [jas_seqent_t](#) v)

Set an element of a sequence.

- static JAS_ATTRIBUTE_PURE [jas_seqent_t](#) * [jas_seq_getref](#) (const [jas_seq_t](#) *seq, [jas_matind_t](#) i)

Get a pointer (i.e., reference) to an element of a sequence.

- static JAS_ATTRIBUTE_PURE [jas_seqent_t](#) [jas_seq_get](#) (const [jas_seq_t](#) *seq, [jas_matind_t](#) i)

Get an element of a sequence.

- static JAS_ATTRIBUTE_PURE [jas_matind_t](#) [jas_seq_start](#) (const [jas_seq_t](#) *seq)

Get the starting index of a sequence.

- static JAS_ATTRIBUTE_PURE [jas_matind_t](#) [jas_seq_end](#) (const [jas_seq_t](#) *seq)

Get the ending index of a sequence.

16.27.1 Detailed Description

Sequence/Matrix Library.

16.28 jas_seq.h

[Go to the documentation of this file.](#)

```
00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00064
00065
00066
00067
00068
00069 #ifndef JAS_SEQ_H
00070 #define JAS_SEQ_H
```

```

00071
00072 /*****
00073  * Includes.
00074  *****/
00075
00076 /* The configuration header file should be included first. */
00077 #include <jasper/jas_config.h> /* IWYU pragma: keep */
00078
00079 #include <jasper/jas_types.h>
00080 #include <jasper/jas_fix.h>
00081 #include <jasper/jas_math.h>
00082
00083 #include <stdio.h>
00084
00085 #ifdef __cplusplus
00086 extern "C" {
00087 #endif
00088
00093
00094 /*****
00095  * Constants.
00096  *****/
00097
00098 /* This matrix is a reference to another matrix. */
00099 #define JAS_MATRIX_REF 0x0001
00100
00101 /*****
00102  * Types.
00103  *****/
00104
00105 /* An element in a sequence. */
00106 typedef jas_fix_t jas_seqent_t;
00107 #define PRIjas_seqent PRIjas_fix
00108
00109 /* An element in a matrix. */
00110 typedef jas_fix_t jas_matent_t;
00111
00112 typedef jas_fix_t jas_matind_t;
00113
00117 typedef struct {
00118
00119     /* Additional state information. */
00120     int flags_;
00121
00122     /* The starting horizontal index. */
00123     jas_matind_t xstart_;
00124
00125     /* The starting vertical index. */
00126     jas_matind_t ystart_;
00127
00128     /* The ending horizontal index. */
00129     jas_matind_t xend_;
00130
00131     /* The ending vertical index. */
00132     jas_matind_t yend_;
00133
00134     /* The number of rows in the matrix. */
00135     jas_matind_t numRows_;
00136
00137     /* The number of columns in the matrix. */
00138     jas_matind_t numcols_;
00139
00140     /* Pointers to the start of each row. */
00141     jas_seqent_t **rows_;
00142
00143     /* The allocated size of the rows array. */
00144     int_fast32_t maxrows_;
00145
00146     /* The matrix data buffer. */
00147     jas_seqent_t *data_;
00148
00149     /* The allocated size of the data array. */
00150     int_fast32_t datasize_;
00151 } jas_matrix_t;
00152
00153
00158 typedef jas_matrix_t jas_seq2d_t;
00159
00164 typedef jas_matrix_t jas_seq_t;
00165
00166 /*****

```

```

00167 * Functions/macros for matrix class.
00168 \*****/
00169
00174 JAS_ATTRIBUTE_PURE
00175 static inline jas_matind_t jas_matrix_numrows(const jas_matrix_t *matrix)
00176 {
00177     return matrix->numrows_;
00178 }
00179
00183 JAS_ATTRIBUTE_PURE
00184 static inline jas_matind_t jas_matrix_numcols(const jas_matrix_t *matrix)
00185 {
00186     return matrix->numcols_;
00187 }
00188
00193 JAS_ATTRIBUTE_PURE
00194 static inline jas_matind_t jas_matrix_size(const jas_matrix_t *matrix)
00195 {
00196     return jas_matrix_numcols(matrix) * jas_matrix_numrows(matrix);
00197 }
00198
00203 JAS_ATTRIBUTE_PURE
00204 static inline bool jas_matrix_empty(const jas_matrix_t *matrix)
00205 {
00206     return jas_matrix_numcols(matrix) == 0 || jas_matrix_numrows(matrix) == 0;
00207 }
00208
00213 JAS_ATTRIBUTE_PURE
00214 static inline jas_segent_t jas_matrix_get(const jas_matrix_t *matrix, jas_matind_t i, jas_matind_t j)
00215 {
00216     assert(i >= 0 && i < matrix->numrows_ && j >= 0 && j < matrix->numcols_);
00217     return matrix->rows_[i][j];
00218 }
00219
00224 static inline void jas_matrix_set(jas_matrix_t *matrix, jas_matind_t i, jas_matind_t j, jas_segent_t v)
00225 {
00226     assert(i >= 0 && i < matrix->numrows_ && j >= 0 && j < matrix->numcols_);
00227     matrix->rows_[i][j] = v;
00228 }
00229
00234 JAS_ATTRIBUTE_PURE
00235 static inline jas_segent_t jas_matrix_getv(const jas_matrix_t *matrix, jas_matind_t i)
00236 {
00237     return matrix->numrows_ == 1
00238         ? matrix->rows_[0][i]
00239         : matrix->rows_[i][0];
00240 }
00241
00246 static inline void jas_matrix_setv(jas_matrix_t *matrix, jas_matind_t i, jas_segent_t v)
00247 {
00248     if (matrix->numrows_ == 1)
00249         matrix->rows_[0][i] = v;
00250     else
00251         matrix->rows_[i][0] = v;
00252 }
00253
00258 JAS_ATTRIBUTE_PURE
00259 static inline jas_segent_t *jas_matrix_getref(const jas_matrix_t *matrix, jas_matind_t i, jas_matind_t j)
00260 {
00261     return &matrix->rows_[i][j];
00262 }
00263
00268 JAS_ATTRIBUTE_PURE
00269 static inline jas_segent_t *jas_matrix_getvref(const jas_matrix_t *matrix, jas_matind_t i)
00270 {
00271     return matrix->numrows_ > 1
00272         ? jas_matrix_getref(matrix, i, 0)
00273         : jas_matrix_getref(matrix, 0, i);
00274 }
00275
00280 JAS_EXPORT
00281 jas_matrix_t *jas_matrix_create(jas_matind_t numrows, jas_matind_t numcols);
00282
00287 JAS_EXPORT
00288 void jas_matrix_destroy(jas_matrix_t *matrix);
00289
00294 JAS_EXPORT
00295 int jas_matrix_resize(jas_matrix_t *matrix, jas_matind_t numrows, jas_matind_t numcols);
00296
00301 JAS_EXPORT
00302 int jas_matrix_output(jas_matrix_t *matrix, FILE *out);

```

```

00303
00308 JAS_EXPORT
00309 int jas_matrix_bindsub(jas_matrix_t *mat0, jas_matrix_t *mat1, jas_matind_t r0,
00310   jas_matind_t c0, jas_matind_t r1, jas_matind_t c1);
00311
00316 static inline int jas_matrix_bindrow(jas_matrix_t *mat0, jas_matrix_t *mat1, jas_matind_t r)
00317 {
00318     return jas_matrix_bindsub(mat0, mat1, r, 0, r, mat1->numcols_ - 1);
00319 }
00320
00325 static inline int jas_matrix_bindcol(jas_matrix_t *mat0, jas_matrix_t *mat1, jas_matind_t c)
00326 {
00327     return jas_matrix_bindsub(mat0, mat1, 0, c, mat1->numrows_ - 1, c);
00328 }
00329
00334 JAS_EXPORT
00335 void jas_matrix_clip(jas_matrix_t *matrix, jas_segent_t minval,
00336   jas_segent_t maxval);
00337
00342 JAS_EXPORT
00343 void jas_matrix_asl(jas_matrix_t *matrix, unsigned n);
00344
00349 JAS_EXPORT
00350 void jas_matrix_asr(jas_matrix_t *matrix, unsigned n);
00351
00356 JAS_EXPORT
00357 void jas_matrix_divpow2(jas_matrix_t *matrix, unsigned n);
00358
00363 JAS_EXPORT
00364 void jas_matrix_setall(jas_matrix_t *matrix, jas_segent_t val);
00365
00370 JAS_ATTRIBUTE_PURE
00371 static inline size_t jas_matrix_rowstep(const jas_matrix_t *matrix)
00372 {
00373     return matrix->numrows_ > 1
00374         ? (size_t)(matrix->rows_[1] - matrix->rows_[0])
00375         : 0u;
00376 }
00377
00382 JAS_ATTRIBUTE_PURE
00383 static inline size_t jas_matrix_step(const jas_matrix_t *matrix)
00384 {
00385     return matrix->numrows_ > 1
00386         ? jas_matrix_rowstep(matrix)
00387         : 1;
00388 }
00389
00394 JAS_EXPORT
00395 int jas_matrix_cmp(jas_matrix_t *mat0, jas_matrix_t *mat1);
00396
00401 JAS_EXPORT
00402 jas_matrix_t *jas_matrix_copy(jas_matrix_t *x);
00403
00408 JAS_EXPORT
00409 jas_matrix_t *jas_matrix_input(FILE *);
00410
00414 JAS_ATTRIBUTE_CONST
00415 static inline jas_segent_t jas_segent_asl(jas_segent_t x, unsigned n)
00416 {
00417     #if 0
00418     #ifdef JAS_ENABLE_32BIT
00419         return jas_least32_asl(x, n);
00420     #else
00421         return jas_fast32_asl(x, n);
00422     #endif
00423     #endif
00424     return jas_fix_asl(x, n);
00425 }
00426
00430 JAS_ATTRIBUTE_CONST
00431 static inline jas_segent_t jas_segent_asr(jas_segent_t x, unsigned n)
00432 {
00433     #if 0
00434     #ifdef JAS_ENABLE_32BIT
00435         return jas_least32_asr(x, n);
00436     #else
00437         return jas_fast32_asr(x, n);
00438     #endif
00439     #endif
00440     return jas_fix_asr(x, n);
00441 }

```



```

00442
00443 /*****
00444  * Functions/macros for 2-D sequence class.
00445  *****/
00446
00451 JAS_EXPORT
00452 jas_seq2d_t *jas_seq2d_copy(jas_seq2d_t *x);
00453
00458 JAS_EXPORT
00459 jas_matrix_t *jas_seq2d_create(jas_matind_t xstart, jas_matind_t ystart,
00460     jas_matind_t xend, jas_matind_t yend);
00461
00466 static inline void jas_seq2d_destroy(jas_seq2d_t *s)
00467 {
00468     jas_matrix_destroy(s);
00469 }
00470
00475 JAS_ATTRIBUTE_PURE
00476 static inline jas_matind_t jas_seq2d_xstart(const jas_seq2d_t *s)
00477 {
00478     return s->xstart_;
00479 }
00480
00485 JAS_ATTRIBUTE_PURE
00486 static inline jas_matind_t jas_seq2d_ystart(const jas_seq2d_t *s)
00487 {
00488     return s->ystart_;
00489 }
00490
00495 JAS_ATTRIBUTE_PURE
00496 static inline jas_matind_t jas_seq2d_xend(const jas_seq2d_t *s)
00497 {
00498     return s->xend_;
00499 }
00500
00505 JAS_ATTRIBUTE_PURE
00506 static inline jas_matind_t jas_seq2d_yend(const jas_seq2d_t *s)
00507 {
00508     return s->yend_;
00509 }
00510
00515 JAS_ATTRIBUTE_PURE
00516 static inline jas_seqent_t *jas_seq2d_getref(const jas_seq2d_t *s, jas_matind_t x, jas_matind_t y)
00517 {
00518     return jas_matrix_getref(s, y - s->ystart_, x - s->xstart_);
00519 }
00520
00525 JAS_ATTRIBUTE_PURE
00526 static inline jas_seqent_t jas_seq2d_get(const jas_seq2d_t *s, jas_matind_t x, jas_matind_t y)
00527 {
00528     return jas_matrix_get(s, y - s->ystart_, x - s->xstart_);
00529 }
00530
00535 JAS_ATTRIBUTE_PURE
00536 static inline size_t jas_seq2d_rowstep(const jas_seq2d_t *s)
00537 {
00538     return jas_matrix_rowstep(s);
00539 }
00540
00545 JAS_ATTRIBUTE_PURE
00546 static inline unsigned jas_seq2d_width(const jas_seq2d_t *s)
00547 {
00548     return (unsigned)(s->xend_ - s->xstart_);
00549 }
00550
00555 JAS_ATTRIBUTE_PURE
00556 static inline unsigned jas_seq2d_height(const jas_seq2d_t *s)
00557 {
00558     return (unsigned)(s->yend_ - s->ystart_);
00559 }
00560
00565 static inline void jas_seq2d_setshift(jas_seq2d_t *s, jas_matind_t x, jas_matind_t y)
00566 {
00567     s->xstart_ = x;
00568     s->ystart_ = y;
00569     s->xend_ = s->xstart_ + s->numcols_;
00570     s->yend_ = s->ystart_ + s->numrows_;
00571 }
00572
00577 JAS_ATTRIBUTE_PURE
00578 static inline jas_matind_t jas_seq2d_size(const jas_seq2d_t *s)

```

```

00579 {
00580     return jas_seq2d_width(s) * jas_seq2d_height(s);
00581 }
00582
00587 JAS_ATTRIBUTE_PURE
00588 static inline bool jas_seq2d_empty(const jas_seq2d_t *s)
00589 {
00590     return jas_seq2d_width(s) == 0 || jas_seq2d_height(s) == 0;
00591 }
00592
00597 JAS_EXPORT
00598 int jas_seq2d_bindsub(jas_matrix_t *s, jas_matrix_t *s1, jas_matind_t xstart,
00599     jas_matind_t ystart, jas_matind_t xend, jas_matind_t yend);
00600
00601 /*****
00602 * Functions/macros for 1-D sequence class.
00603 *****/
00604
00609 static inline jas_seq_t *jas_seq_create(jas_matind_t start, jas_matind_t end)
00610 {
00611     return jas_seq2d_create(start, 0, end, 1);
00612 }
00613
00618 static inline void jas_seq_destroy(jas_seq_t *seq)
00619 {
00620     jas_seq2d_destroy(seq);
00621 }
00622
00627 static inline void jas_seq_set(jas_seq_t *seq, jas_matind_t i, jas_segent_t v)
00628 {
00629     seq->rows_[0][i - seq->xstart_] = v;
00630 }
00631
00636 JAS_ATTRIBUTE_PURE
00637 static inline jas_segent_t *jas_seq_getref(const jas_seq_t *seq, jas_matind_t i)
00638 {
00639     return &seq->rows_[0][i - seq->xstart_];
00640 }
00641
00646 JAS_ATTRIBUTE_PURE
00647 static inline jas_segent_t jas_seq_get(const jas_seq_t *seq, jas_matind_t i)
00648 {
00649     return seq->rows_[0][i - seq->xstart_];
00650 }
00651
00656 JAS_ATTRIBUTE_PURE
00657 static inline jas_matind_t jas_seq_start(const jas_seq_t *seq)
00658 {
00659     return seq->xstart_;
00660 }
00661
00666 JAS_ATTRIBUTE_PURE
00667 static inline jas_matind_t jas_seq_end(const jas_seq_t *seq)
00668 {
00669     return seq->xend_;
00670 }
00671
00675
00676 #ifdef __cplusplus
00677 }
00678 #endif
00679
00680 #endif

```

16.29 jas_stream.h File Reference

I/O Stream Class.

```

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

```

Classes

- struct [jas_stream_t](#)
I/O stream object.

Macros

- #define [jas_stream_eof](#)(stream)
Get the EOF indicator for a stream.
- #define [jas_stream_error](#)(stream)
Get the error indicator for a stream.
- #define [jas_stream_clearerr](#)(stream)
Clear the error indicator for a stream.
- #define [jas_stream_getrwlmit](#)(stream)
Get the read/write limit for a stream.
- #define [jas_stream_getrwcoun](#)(stream)
Get the read/write count for a stream.
- #define [jas_stream_getc](#)(stream)
jas_stream_getc Read a character from a stream.
- #define [jas_stream_putc](#)(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_EXPORT [jas_stream_t](#) * [jas_stream_fopen](#) (const char *filename, const char *mode)
Open a file as a stream.
- JAS_EXPORT [jas_stream_t](#) * [jas_stream_memopen](#) (char *buffer, size_t buffer_size)
Open a memory buffer as a stream.
- JAS_DEPRECATED JAS_EXPORT [jas_stream_t](#) * [jas_stream_memopen2](#) (char *buffer, size_t buffer_size)
Do not use.
- JAS_EXPORT [jas_stream_t](#) * [jas_stream_fdopen](#) (int fd, const char *mode)
Open a file descriptor as a stream.
- JAS_EXPORT [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_EXPORT [jas_stream_t](#) * [jas_stream_tmpfile](#) (void)
Open a temporary file as a stream.
- JAS_EXPORT int [jas_stream_close](#) ([jas_stream_t](#) *stream)
Close a stream.
- JAS_EXPORT long [jas_stream_setrwlmit](#) ([jas_stream_t](#) *stream, long rwlmit)
Set the read/write limit for a stream.
- JAS_EXPORT long [jas_stream_setrwcoun](#) ([jas_stream_t](#) *stream, long rw_count)
Set the read/write count for a stream.
- JAS_EXPORT size_t [jas_stream_read](#) ([jas_stream_t](#) *stream, void *buffer, size_t count)
Read characters from a stream into a buffer.

- JAS_EXPORT 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_EXPORT size_t [jas_stream_write](#) ([jas_stream_t](#) *stream, const void *buffer, size_t count)
Write characters from a buffer to a stream.
- JAS_EXPORT int [jas_stream_printf](#) ([jas_stream_t](#) *stream, const char *format,...)
Write formatted output to a stream.
- JAS_EXPORT int [jas_stream_puts](#) ([jas_stream_t](#) *stream, const char *s)
Write a string to a stream.
- JAS_EXPORT char * [jas_stream_gets](#) ([jas_stream_t](#) *stream, char *buffer, int buffer_size)
Read a line of input from a stream.
- JAS_EXPORT int [jas_stream_ungetc](#) ([jas_stream_t](#) *stream, int c)
Put a character back on a stream.
- JAS_EXPORT JAS_ATTRIBUTE_PURE int [jas_stream_isseekable](#) ([jas_stream_t](#) *stream)
Determine if stream supports seeking.
- JAS_EXPORT long [jas_stream_seek](#) ([jas_stream_t](#) *stream, long offset, int origin)
Set the current position within the stream.
- JAS_EXPORT long [jas_stream_tell](#) ([jas_stream_t](#) *stream)
Get the current position within the stream.
- JAS_EXPORT int [jas_stream_rewind](#) ([jas_stream_t](#) *stream)
Seek to the beginning of a stream.
- JAS_EXPORT int [jas_stream_flush](#) ([jas_stream_t](#) *stream)
Flush any pending output to a stream.
- JAS_EXPORT int [jas_stream_copy](#) ([jas_stream_t](#) *destination, [jas_stream_t](#) *source, ssize_t count)
Copy data from one stream to another.
- JAS_EXPORT int [jas_stream_display](#) ([jas_stream_t](#) *stream, FILE *fp, int count)
Print a hex dump of data read from a stream.
- JAS_EXPORT ssize_t [jas_stream_gobble](#) ([jas_stream_t](#) *stream, size_t count)
Consume (i.e., discard) characters from stream.
- JAS_EXPORT ssize_t [jas_stream_pad](#) ([jas_stream_t](#) *stream, size_t count, int value)
Write a fill character multiple times to a stream.
- JAS_EXPORT long [jas_stream_length](#) ([jas_stream_t](#) *stream)
Get the size of the file associated with the specified stream.

16.29.1 Detailed Description

I/O Stream Class.

16.30 jas_stream.h

[Go to the documentation of this file.](#)

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2003 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00064
00065 #ifndef JAS_STREAM_H
00066 #define JAS_STREAM_H
00067
00068
00069 /*****
00070  * Includes.
00071  *****/
00072
00073 /* The configuration header file should be included first. */
00074 #include <jasper/jas_config.h> /* IWYU pragma: export */
00075
00076 #include <stdio.h>

```

```

00080 #if defined(JAS_HAVE_FCNTL_H)
00081 #include <fcntl.h>
00082 #endif
00083 #include <jasper/jas_types.h>
00084
00085 #ifdef __cplusplus
00086 extern "C" {
00087 #endif
00088
00093
00094 /*****
00095  * Constants.
00096  *****/
00097
00098 /* On most UNIX systems, we probably need to define O_BINARY ourselves. */
00099 #ifndef O_BINARY
00100 #define O_BINARY      0
00101 #endif
00102
00103 /*
00104  * Stream open flags.
00105  */
00106
00107 /* The stream was opened for reading. */
00108 #define JAS_STREAM_READ 0x0001
00109 /* The stream was opened for writing. */
00110 #define JAS_STREAM_WRITE 0x0002
00111 /* The stream was opened for appending. */
00112 #define JAS_STREAM_APPEND 0x0004
00113 /* The stream was opened in binary mode. */
00114 #define JAS_STREAM_BINARY 0x0008
00115 /* The stream should be created/truncated. */
00116 #define JAS_STREAM_CREATE 0x0010
00117
00118 /*
00119  * Stream buffering flags.
00120  */
00121
00122 /* The stream is unbuffered. */
00123 #define JAS_STREAM_UNBUF 0x0000
00124 /* The stream is line buffered. */
00125 #define JAS_STREAM_LINEBUF 0x0001
00126 /* The stream is fully buffered. */
00127 #define JAS_STREAM_FULLBUF 0x0002
00128 /* The buffering mode mask. */
00129 #define JAS_STREAM_BUFMODEMASK 0x000f
00130
00131 /* The memory associated with the buffer needs to be deallocated when the
00132  stream is destroyed. */
00133 #define JAS_STREAM_FREEBUF 0x0008
00134 /* The buffer is currently being used for reading. */
00135 #define JAS_STREAM_RDBUF 0x0010
00136 /* The buffer is currently being used for writing. */
00137 #define JAS_STREAM_WRBUF 0x0020
00138
00139 /*
00140  * Stream error flags.
00141  */
00142
00143 /* The end-of-file has been encountered (on reading). */
00144 #define JAS_STREAM_EOF 0x0001
00145 /* An I/O error has been encountered on the stream. */
00146 #define JAS_STREAM_ERR 0x0002
00147 /* The read/write limit has been exceeded. */
00148 #define JAS_STREAM_RWLIMIT 0x0004
00149 /* The error mask. */
00150 #define JAS_STREAM_ERRMASK \
00151     (JAS_STREAM_EOF | JAS_STREAM_ERR | JAS_STREAM_RWLIMIT)
00152
00153 /*
00154  * Other miscellaneous constants.
00155  */
00156
00157 /* The default buffer size (for fully-buffered operation). */
00158 #define JAS_STREAM_BUFSIZE 8192
00159 /* The default permission mask for file creation. */
00160 #define JAS_STREAM_PERMS 0666
00161
00162 /* The maximum number of characters that can always be put back on a stream. */
00163 #define JAS_STREAM_MAXPUTBACK 16
00164

```

```

00165 /*****
00166  * Types.
00167  *****/
00168 /*
00169  * Generic file object.
00170  */
00171 */
00172
00173 typedef void jas_stream_obj_t;
00174
00175 /*
00176  * Generic file object operations.
00177  */
00178
00179 typedef struct {
00180
00181     /* Read characters from a file object. */
00182     ssize_t (*read_)(jas_stream_obj_t *obj, char *buf, size_t cnt);
00183
00184     /* Write characters to a file object. */
00185     ssize_t (*write_)(jas_stream_obj_t *obj, const char *buf, size_t cnt);
00186
00187     /* Set the position for a file object. */
00188     long (*seek_)(jas_stream_obj_t *obj, long offset, int origin);
00189
00190     /* Close a file object. */
00191     int (*close_)(jas_stream_obj_t *obj);
00192 } jas_stream_ops_t;
00193
00194
00205 typedef struct {
00206
00207     /* The mode in which the stream was opened. */
00208     int openmode_;
00209
00210     /* The buffering mode. */
00211     int bufmode_;
00212
00213     /* The stream status. */
00214     int flags_;
00215
00216     /* The start of the buffer area to use for reading/writing. */
00217     jas_uchar *bufbase_;
00218
00219     /* The start of the buffer area excluding the extra initial space for
00220      character putback. */
00221     jas_uchar *bufstart_;
00222
00223     /* The buffer size. */
00224     int bufsize_;
00225
00226     /* The current position in the buffer. */
00227     jas_uchar *ptr_;
00228
00229     /* The number of characters that must be read/written before
00230      the buffer needs to be filled/flushed. */
00231     int cnt_;
00232
00233     /* A trivial buffer to be used for unbuffered operation. */
00234     jas_uchar tinybuf_[JAS_STREAM_MAXPUTBACK + 1];
00235
00236     /* The operations for the underlying stream file object. */
00237     const jas_stream_ops_t *ops_;
00238
00239     /* The underlying stream file object. */
00240     jas_stream_obj_t *obj_;
00241
00242     /* The number of characters read/written. */
00243     long rwcnt_;
00244
00245     /* The maximum number of characters that may be read/written. */
00246     long rwlimit_;
00247 } jas_stream_t;
00248
00249
00250 /*
00251  * Regular file object.
00252  */
00253 */
00254
00255 /*

```

```

00256  * File descriptor file object.
00257  */
00258  typedef struct {
00259      int fd;
00260      int flags;
00261      #if defined(JAS_WASI_LIBC)
00262      #define L_tmpnam 4096
00263      #endif
00264      char pathname[L_tmpnam + 1];
00265  } jas_stream_fileobj_t;
00266
00267  /* Delete underlying file object upon stream close. */
00268  #define JAS_STREAM_FILEOBJ_DELCLOSE 0x01
00269  /* Do not close underlying file object upon stream close. */
00270  #define JAS_STREAM_FILEOBJ_NOCLOSE 0x02
00271
00272  /*
00273   * Memory file object.
00274   */
00275
00276  typedef struct {
00277
00278      /* The data associated with this file. */
00279      jas_uchar *buf_;
00280
00281      /* The allocated size of the buffer for holding file data. */
00282      size_t bufsize_;
00283
00284      /* The length of the file. */
00285      size_t len_;
00286
00287      /* The seek position. */
00288      size_t pos_;
00289
00290      /* Is the buffer growable? */
00291      int growable_;
00292
00293      /* Was the buffer allocated internally? */
00294      int myalloc_;
00295
00296  } jas_stream_memobj_t;
00297
00298  /*****
00299   * Macros/functions for opening and closing streams.
00300   \*****/
00301
00316  JAS_EXPORT
00317  jas_stream_t *jas_stream_fopen(const char *filename, const char *mode);
00318
00343  JAS_EXPORT
00344  jas_stream_t *jas_stream_memopen(char *buffer, size_t buffer_size);
00345
00354  JAS_DEPRECATED
00355  JAS_EXPORT
00356  jas_stream_t *jas_stream_memopen2(char *buffer, size_t buffer_size);
00357
00372  JAS_EXPORT
00373  jas_stream_t *jas_stream_fdopen(int fd, const char *mode);
00374
00397  JAS_EXPORT
00398  jas_stream_t *jas_stream_freopen(const char *path, const char *mode, FILE *fp);
00399
00414  JAS_EXPORT
00415  jas_stream_t *jas_stream_tmpfile(void);
00416
00435  JAS_EXPORT
00436  int jas_stream_close(jas_stream_t *stream);
00437
00438  /*****
00439   * Macros/functions for getting/setting the stream state.
00440   \*****/
00441
00452  #define jas_stream_eof(stream) \
00453      (((stream)->flags_ & JAS_STREAM_EOF) != 0)
00454
00466  #define jas_stream_error(stream) \
00467      (((stream)->flags_ & JAS_STREAM_ERR) != 0)
00468
00478  #define jas_stream_clearerr(stream) \
00479      ((stream)->flags_ &= ~(JAS_STREAM_ERR | JAS_STREAM_EOF))
00480

```



```

00491 #define jas_stream_getrwlimit(stream) \
00492     (((const jas_stream_t *) (stream))->rwlimit_)
00493
00509 JAS_EXPORT long jas_stream_setrwlimit(jas_stream_t *stream, long rwlimit);
00510
00521 #define jas_stream_getrwcountrwcount(stream) \
00522     (((const jas_stream_t *) (stream))->rwcnt_)
00523
00537 JAS_EXPORT
00538 long jas_stream_setrwcountrwcount(jas_stream_t *stream, long rw_count);
00539
00540 /*****
00541  * Macros/functions for I/O.
00542  *****/
00543
00544 /* Read a character from a stream. */
00545 #ifndef NDEBBUG
00557 #define jas_stream_getc(stream) jas_stream_getc_func(stream)
00558 #else
00559 #define jas_stream_getc(stream) jas_stream_getc_macro(stream)
00560 #endif
00561
00562 /* Write a character to a stream. */
00563 #ifndef NDEBBUG
00578 #define jas_stream_putc(stream, c) jas_stream_putc_func(stream, c)
00579 #else
00580 #define jas_stream_putc(stream, c) jas_stream_putc_macro(stream, c)
00581 #endif
00582
00621 JAS_EXPORT
00622 size_t jas_stream_read(jas_stream_t *stream, void *buffer, size_t count);
00623
00649 JAS_EXPORT
00650 unsigned jas_stream_peek(jas_stream_t *stream, void *buffer, size_t count);
00651
00681 JAS_EXPORT
00682 size_t jas_stream_write(jas_stream_t *stream, const void *buffer,
00683     size_t count);
00684
00706 JAS_EXPORT
00707 int jas_stream_printf(jas_stream_t *stream, const char *format, ...);
00708
00726 JAS_EXPORT
00727 int jas_stream_puts(jas_stream_t *stream, const char *s);
00728
00752 JAS_EXPORT
00753 char *jas_stream_gets(jas_stream_t *stream, char *buffer, int buffer_size);
00754
00773 #define jas_stream_peekc(stream) \
00774     (((stream)->cnt_ <= 0) ? jas_stream_fillbuf(stream, 0) : \
00775     ((int) (*(stream)->ptr_)))
00776
00800 JAS_EXPORT
00801 int jas_stream_ungetc(jas_stream_t *stream, int c);
00802
00803 /*****
00804  * Macros/functions for getting/setting the stream position.
00805  *****/
00806
00822 JAS_EXPORT
00823 JAS_ATTRIBUTE_PURE
00824 int jas_stream_isseekable(jas_stream_t *stream);
00825
00845 JAS_EXPORT
00846 long jas_stream_seek(jas_stream_t *stream, long offset, int origin);
00847
00863 JAS_EXPORT
00864 long jas_stream_tell(jas_stream_t *stream);
00865
00883 JAS_EXPORT
00884 int jas_stream_rewind(jas_stream_t *stream);
00885
00886 /*****
00887  * Macros/functions for flushing.
00888  *****/
00889
00905 JAS_EXPORT
00906 int jas_stream_flush(jas_stream_t *stream);
00907
00908 /*****
00909  * Miscellaneous macros/functions.

```

```

00910 \*****/
00911
00941 JAS_EXPORT
00942 int jas_stream_copy(jas_stream_t *destination, jas_stream_t *source,
00943     ssize_t count);
00944
00967 JAS_EXPORT
00968 int jas_stream_display(jas_stream_t *stream, FILE *fp, int count);
00969
00989 JAS_EXPORT
00990 ssize_t jas_stream_gobble(jas_stream_t *stream, size_t count);
00991
01013 JAS_EXPORT
01014 ssize_t jas_stream_pad(jas_stream_t *stream, size_t count, int value);
01015
01035 JAS_EXPORT
01036 long jas_stream_length(jas_stream_t *stream);
01037
01038 /*****\
01039  * Internal functions.
01040  \*****/
01041
01042 /* The following functions are for internal use only!  If you call them
01043 directly, you will die a horrible, miserable, and painful death! */
01044
01045 /* These prototypes need to be here for the sake of the stream_getc and
01046 stream_putc macros. */
01047 /* Library users must not invoke these functions directly. */
01048 JAS_EXPORT int jas_stream_fillbuf(jas_stream_t *stream, int getflag);
01049 JAS_EXPORT int jas_stream_flushbuf(jas_stream_t *stream, int c);
01050 JAS_EXPORT int jas_stream_getc_func(jas_stream_t *stream);
01051 JAS_EXPORT int jas_stream_putc_func(jas_stream_t *stream, int c);
01052
01053 /* Read a character from a stream. */
01054 static inline int jas_stream_getc2(jas_stream_t *stream)
01055 {
01056     if (--stream->cnt_ < 0)
01057         return jas_stream_fillbuf(stream, 1);
01058
01059     ++stream->rwcnt_;
01060     return (int) (*stream->ptr_++);
01061 }
01062
01063 static inline int jas_stream_getc_macro(jas_stream_t *stream)
01064 {
01065     if (stream->flags_ & (JAS_STREAM_ERR | JAS_STREAM_EOF | JAS_STREAM_RWLIMIT))
01066         return EOF;
01067
01068     if (stream->rwlimit_ >= 0 && stream->rwcnt_ >= stream->rwlimit_) {
01069         stream->flags_ |= JAS_STREAM_RWLIMIT;
01070         return EOF;
01071     }
01072
01073     return jas_stream_getc2(stream);
01074 }
01075
01076 /* Write a character to a stream. */
01077 static inline int jas_stream_putc2(jas_stream_t *stream, jas_uchar c)
01078 {
01079     stream->bufmode_ |= JAS_STREAM_WRBUF;
01080
01081     if (--stream->cnt_ < 0)
01082         return jas_stream_flushbuf(stream, c);
01083     else {
01084         ++stream->rwcnt_;
01085         return (int) (*stream->ptr_++ = c);
01086     }
01087 }
01088
01089 static inline int jas_stream_putc_macro(jas_stream_t *stream, jas_uchar c)
01090 {
01091     if (stream->flags_ & (JAS_STREAM_ERR | JAS_STREAM_EOF | JAS_STREAM_RWLIMIT))
01092         return EOF;
01093
01094     if (stream->rwlimit_ >= 0 && stream->rwcnt_ >= stream->rwlimit_) {
01095         stream->flags_ |= JAS_STREAM_RWLIMIT;
01096         return EOF;
01097     }
01098
01099     return jas_stream_putc2(stream, c);
01100 }

```

```

01101
01105
01106 #ifdef __cplusplus
01107 }
01108 #endif
01109
01110 #endif

```

16.31 jas_string.h File Reference

String Library.

```

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

```

Functions

- JAS_EXPORT char * [jas_strdup](#) (const char *)
Create a copy of a null-terminated string.
- JAS_EXPORT char * [jas_strtok](#) (char *str, const char *delim, char **saveptr)
Extract tokens from a string.
- JAS_EXPORT int [jas_stringtokenize](#) (const char *string, const char *delim, char ***tokens_buf, size_t *max_tokens_buf, size_t *num_tokens_buf)
Split a string into tokens based on specified delimiters.

16.31.1 Detailed Description

String Library.

16.32 jas_string.h

[Go to the documentation of this file.](#)

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,

```

```

00022 * publish, distribute, and/or sell copies of the Software, and to permit
00023 * persons to whom the Software is furnished to do so, subject to the
00024 * following conditions:
00025 *
00026 * 1. The above copyright notices and this permission notice (which
00027 * includes the disclaimer below) shall be included in all copies or
00028 * substantial portions of the Software.
00029 *
00030 * 2. The name of a copyright holder shall not be used to endorse or
00031 * promote products derived from the Software without specific prior
00032 * written permission.
00033 *
00034 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060 *
00061 * __END_OF_JASPER_LICENSE__
00062 */
00063
00064
00065 #ifndef JAS_STRING_H
00066 #define JAS_STRING_H
00067
00068
00069 /*****
00070 * Includes.
00071 *****/
00072 /* The configuration header file should be included first. */
00073 #include <jasper/jas_config.h>
00074
00075 #include <jasper/jas_types.h>
00076
00077 #ifdef __cplusplus
00078 extern "C" {
00079 #endif
00080
00081 /*****
00082 * Functions.
00083 *****/
00084 #ifdef JAS_EXPORT
00085 char *jas_strdup(const char *);
00086
00087 JAS_EXPORT
00088 char *jas_strtok(char *str, const char *delim, char **saveptr);
00089
00090 JAS_EXPORT
00091 int jas_stringtokenize(const char *string, const char *delim,
00092 char ***tokens_buf, size_t *max_tokens_buf, size_t *num_tokens_buf);
00093
00094 #ifdef __cplusplus
00095 }
00096 #endif
00097 #endif

```

16.33 jas_thread.h File Reference

Threads.

```
#include <jasper/jas_config.h>
#include "jasper/jas_compiler.h"
#include "jasper/jas_types.h"
#include <stdlib.h>
#include <assert.h>
#include <threads.h>
#include <stdatomic.h>
```

16.33.1 Detailed Description

Threads.

16.34 jas_thread.h

[Go to the documentation of this file.](#)

```
00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2003 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
```

```

00044 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060 *
00061 * __END_OF_JASPER_LICENSE__
00062 */
00063
00068
00069 #ifndef JAS_THREAD_H
00070 #define JAS_THREAD_H
00071
00072 /*****
00073  * Includes
00074  *****/
00075
00076 /* The configuration header file should be included first. */
00077 #include <jasper/jas_config.h>
00078
00079 #include "jasper/jas_compiler.h"
00080 #include "jasper/jas_types.h"
00081
00082 #if defined(JAS_THREADS)
00083
00084 #include <stdlib.h>
00085 #include <assert.h>
00086
00087 #if defined(JAS_THREADS_C11)
00088 #include <threads.h>
00089 #include <stdatomic.h>
00090 #elif defined(JAS_THREADS_PTHREAD)
00091 #include <pthread.h>
00092 #include <sched.h>
00093 #elif defined(JAS_THREADS_WIN32)
00094 #include <process.h>
00095 #include <windows.h>
00096 #include <processthreadsapi.h>
00097 #endif
00098
00099 #endif
00100
00101 /*****
00102  *****/
00103
00104 #ifdef __cplusplus
00105 extern "C" {
00106 #endif
00107
00113
00114 #if defined(JAS_THREADS)
00115 #if defined(JAS_FOR_INTERNAL_USE_ONLY) || defined(JAS_FOR_JASPER_APP_USE_ONLY)
00116
00117 /*****
00118  * Types
00119  *****/
00120
00121 #if defined(JAS_THREADS_C11)
00122 #   define JAS_THREADS_IMPL "C11"
00123 #   define JAS_USE_SPINLOCK
00124 #elif defined(JAS_THREADS_PTHREAD)
00125 #   define JAS_THREADS_IMPL "PTHREAD"
00126 #   undef JAS_USE_SPINLOCK
00127 #elif defined(JAS_THREADS_WIN32)
00128 #   define JAS_THREADS_IMPL "WIN32"
00129 #   define JAS_USE_SPINLOCK
00130 #endif
00131
00132 /*****
00133  * Spinlock

```

```

00134 \*****/
00135
00137 #if defined(JAS_THREADS_C11)
00138 #     define JAS_USE_SPINLOCK
00139 typedef struct {
00140     atomic_flag flag;
00141 } jas_spinlock_t;
00142 #elif defined(JAS_THREADS_PTHREAD)
00143 /* There is no pthread_spinlock_t type on MacOS. */
00144 #     undef JAS_USE_SPINLOCK
00145 #elif defined(JAS_THREADS_WIN32)
00146 #     define JAS_USE_SPINLOCK
00147 typedef struct {
00148     LONG flag;
00149 } jas_spinlock_t;
00150 #endif
00151
00153 #if defined(JAS_THREADS_C11)
00154 #define JAS_SPINLOCK_INITIALIZER {ATOMIC_FLAG_INIT}
00155 #elif defined(JAS_THREADS_PTHREAD)
00156 /* There is no pthread_spinlock_t type on MacOS. */
00157 #elif defined(JAS_THREADS_WIN32)
00158 #define JAS_SPINLOCK_INITIALIZER {0}
00159 #endif
00160
00161 /*****\
00162 * Basic Mutex
00163 \*****/
00164
00166 #if defined(JAS_THREADS_C11)
00167 typedef mtx_t jas_basicmutex_t;
00168 #elif defined(JAS_THREADS_PTHREAD)
00169 typedef pthread_mutex_t jas_basicmutex_t;
00170 #elif defined(JAS_THREADS_WIN32)
00171 typedef CRITICAL_SECTION jas_basicmutex_t;
00172 #endif
00173
00175 #if defined(JAS_THREADS_C11)
00176 #undef JAS_BASICMUTEX_INITIALIZER
00177 #elif defined(JAS_THREADS_PTHREAD)
00178 #define JAS_BASICMUTEX_INITIALIZER PTHREAD_MUTEX_INITIALIZER
00179 #elif defined(JAS_THREADS_WIN32)
00180 #define JAS_BASICMUTEX_INITIALIZER
00181 #endif
00182
00183 /*****\
00184 * Mutex (Allowing Static Initialization)
00185 \*****/
00186
00187 #if defined(JAS_USE_SPINLOCK)
00188 #     define jas_mutex_t jas_spinlock_t
00189 #     define JAS_MUTEX_INITIALIZER JAS_SPINLOCK_INITIALIZER
00190 #     define jas_mutex_init jas_spinlock_init
00191 #     define jas_mutex_cleanup jas_spinlock_cleanup
00192 #     define jas_mutex_lock jas_spinlock_lock
00193 #     define jas_mutex_unlock jas_spinlock_unlock
00194 #else
00195 #     define jas_mutex_t jas_basicmutex_t
00196 #     define JAS_MUTEX_INITIALIZER JAS_BASICMUTEX_INITIALIZER
00197 #     define jas_mutex_init jas_basicmutex_init
00198 #     define jas_mutex_cleanup jas_basicmutex_cleanup
00199 #     define jas_mutex_lock jas_basicmutex_lock
00200 #     define jas_mutex_unlock jas_basicmutex_unlock
00201 #endif
00202
00203 /*****\
00204 * Once Flag
00205 \*****/
00206
00208 #if defined(JAS_THREADS_C11)
00209 typedef once_flag jas_once_flag_t;
00210 #elif defined(JAS_THREADS_PTHREAD)
00211 typedef pthread_once_t jas_once_flag_t;
00212 #elif defined(JAS_THREADS_WIN32)
00213 typedef struct {
00214     volatile LONG status;
00215 } jas_once_flag_t;
00216 #endif
00217
00219 #if defined(JAS_THREADS_C11)
00220 #     define JAS_ONCE_FLAG_INIT ONCE_FLAG_INIT

```



```

00327         return 0;
00328 #elif defined(JAS_THREADS_PTHREAD)
00329     JAS_UNUSED(mtx);
00330     abort();
00331     return -1;
00332 #elif defined(JAS_THREADS_WIN32)
00333     JAS_UNUSED(mtx);
00334     return 0;
00335 #endif
00336 }
00337
00349 static inline int jas_spinlock_lock(jas_spinlock_t *mtx)
00350 {
00351     assert(mtx);
00352 #if defined(JAS_THREADS_C11)
00353     while (atomic_flag_test_and_set(&mtx->flag)) {}
00354     return 0;
00355 #elif defined(JAS_THREADS_PTHREAD)
00356     JAS_UNUSED(mtx);
00357     abort();
00358     return -1;
00359 #elif defined(JAS_THREADS_WIN32)
00360     while (InterlockedCompareExchange(&mtx->flag, 1, 0)) {}
00361     return 0;
00362 #endif
00363 }
00364
00376 static inline int jas_spinlock_unlock(jas_spinlock_t *mtx)
00377 {
00378     assert(mtx);
00379 #if defined(JAS_THREADS_C11)
00380     atomic_flag_clear(&mtx->flag);
00381     return 0;
00382 #elif defined(JAS_THREADS_PTHREAD)
00383     JAS_UNUSED(mtx);
00384     abort();
00385     return -1;
00386 #elif defined(JAS_THREADS_WIN32)
00387     InterlockedExchange(&mtx->flag, 0);
00388     return 0;
00389 #endif
00390 }
00391
00392 #endif
00393
00394 /*****\
00395  * Basic Mutex
00396  \*****/
00397
00398 /* For internal use only. */
00399 static inline int jas_basicmutex_init(jas_basicmutex_t *mtx)
00400 {
00401     assert(mtx);
00402 #if defined(JAS_THREADS_C11)
00403     return mtx_init(mtx, mtx_plain) == thrd_success ? 0 : -1;
00404 #elif defined(JAS_THREADS_PTHREAD)
00405     return pthread_mutex_init(mtx, 0);
00406 #elif defined(JAS_THREADS_WIN32)
00407     InitializeCriticalSection(mtx);
00408     return 0;
00409 #endif
00410 }
00411
00412 /* For internal use only. */
00413 static inline int jas_basicmutex_cleanup(jas_basicmutex_t *mtx)
00414 {
00415     assert(mtx);
00416 #if defined(JAS_THREADS_C11)
00417     mtx_destroy(mtx);
00418     return 0;
00419 #elif defined(JAS_THREADS_PTHREAD)
00420     return pthread_mutex_destroy(mtx);
00421 #elif defined(JAS_THREADS_WIN32)
00422     DeleteCriticalSection(mtx);
00423     return 0;
00424 #endif
00425 }
00426
00427 /* For internal use only. */
00428 static inline int jas_basicmutex_lock(jas_basicmutex_t *mtx)
00429 {

```

```

00430         assert(mtx);
00431 #if defined(JAS_THREADS_C11)
00432         return mtx_lock(mtx);
00433 #elif defined(JAS_THREADS_PTHREAD)
00434         return pthread_mutex_lock(mtx);
00435 #elif defined(JAS_THREADS_WIN32)
00436         EnterCriticalSection(mtx);
00437         return 0;
00438 #endif
00439 }
00440
00441 /* For internal use only. */
00442 static inline int jas_basicmutex_unlock(jas_basicmutex_t *mtx)
00443 {
00444     assert(mtx);
00445 #if defined(JAS_THREADS_C11)
00446     return mtx_unlock(mtx);
00447 #elif defined(JAS_THREADS_PTHREAD)
00448     return pthread_mutex_unlock(mtx);
00449 #elif defined(JAS_THREADS_WIN32)
00450     LeaveCriticalSection(mtx);
00451     return 0;
00452 #endif
00453 }
00454
00455 /*****
00456  * Thread-Specific Storage (TSS)
00457  *****/
00458
00472 static inline
00473 int jas_tss_create(jas_tss_t *tss, void (*destructor)(void *))
00474 {
00475     assert(tss);
00476 #if defined(JAS_THREADS_C11)
00477     return tss_create(tss, destructor) == thrd_success ? 0 : -1;
00478 #elif defined(JAS_THREADS_PTHREAD)
00479     return pthread_key_create(tss, destructor);
00480 #elif defined(JAS_THREADS_WIN32)
00481     if (destructor) {
00482         return -1;
00483     }
00484     DWORD id;
00485     if ((id = TlsAlloc()) == TLS_OUT_OF_INDEXES) {
00486         return -2;
00487     }
00488     *tss = id;
00489     return 0;
00490 #endif
00491 }
00492
00503 static inline
00504 void jas_tss_delete(jas_tss_t tss)
00505 {
00506 #if defined(JAS_THREADS_C11)
00507     tss_delete(tss);
00508 #elif defined(JAS_THREADS_PTHREAD)
00509     pthread_key_delete(tss);
00510 #elif defined(JAS_THREADS_WIN32)
00511     TlsFree(tss);
00512 #endif
00513 }
00514
00524 static inline
00525 void *jas_tss_get(jas_tss_t tss)
00526 {
00527 #if defined(JAS_THREADS_C11)
00528     return tss_get(tss);
00529 #elif defined(JAS_THREADS_PTHREAD)
00530     return pthread_getspecific(tss);
00531 #elif defined(JAS_THREADS_WIN32)
00532     return TlsGetValue(tss);
00533 #endif
00534 }
00535
00546 static inline
00547 int jas_tss_set(jas_tss_t tss, void *value)
00548 {
00549 #if defined(JAS_THREADS_C11)
00550     return tss_set(tss, value) == thrd_success ? 0 : -1;
00551 #elif defined(JAS_THREADS_PTHREAD)
00552     return pthread_setspecific(tss, value);

```

```

00553 #elif defined(JAS_THREADS_WIN32)
00554     return TlsSetValue(tss, value) ? 0 : -1;
00555 #endif
00556 }
00557
00558 /*****
00559  * Once Flag
00560  *****/
00561
00572 static inline int jas_call_once(jas_once_flag_t *flag, void (*func)(void))
00573 {
00574     assert(flag);
00575     assert(func);
00576     #if defined(JAS_THREADS_C11)
00577         call_once(flag, func);
00578         return 0;
00579     #elif defined(JAS_THREADS_PTHREAD)
00580         return pthread_once(flag, func);
00581     #elif defined(JAS_THREADS_WIN32)
00582         if (InterlockedCompareExchange(&flag->status, 1, 0) == 0) {
00583             (func)();
00584             InterlockedExchange(&flag->status, 2);
00585         } else {
00586             while (flag->status == 1) {
00587                 /* Perform a busy wait. This is ugly. */
00588                 /* Yield processor. */
00589                 SwitchToThread();
00590             }
00591         }
00592         return 0;
00593     #endif
00594 }
00595
00596 /*****
00597  * Threads
00598  *****/
00599
00600 #if defined(JAS_FOR_INTERNAL_USE_ONLY) || defined(JAS_FOR_JASPER_APP_USE_ONLY)
00601
00602 #if defined(JAS_THREADS_PTHREAD)
00603 static void *thread_func_wrapper(void *thread_ptr)
00604 {
00605     jas_thread_t *thread = JAS_CAST(jas_thread_t *, thread_ptr);
00606     int result = (thread->func)(thread->arg);
00607     thread->result = result;
00608     return thread;
00609 }
00610 #elif defined(JAS_THREADS_WIN32)
00611 static unsigned __stdcall thread_func_wrapper(void *thread_ptr)
00612 {
00613     jas_thread_t *thread = JAS_CAST(jas_thread_t *, thread_ptr);
00614     int result = (thread->func)(thread->arg);
00615     return JAS_CAST(unsigned, result);
00616 }
00617 #endif
00618
00627 static inline
00628 int jas_thread_compare(jas_thread_id_t x, jas_thread_id_t y)
00629 {
00630     #if defined(JAS_THREADS_C11)
00631         return thrd_equal(x, y);
00632     #elif defined(JAS_THREADS_PTHREAD)
00633         return pthread_equal(x, y);
00634     #elif defined(JAS_THREADS_WIN32)
00635         return GetThreadId(x) == GetThreadId(y);
00636     #endif
00637 }
00638
00647 static inline
00648 int jas_thread_create(jas_thread_t *thread, int (*func)(void *), void *arg)
00649 {
00650     assert(thread);
00651     assert(func);
00652     #if defined(JAS_THREADS_C11)
00653         return thrd_create(thread, func, arg) == thrd_success ? 0 : -1;
00654     #elif defined(JAS_THREADS_PTHREAD)
00655         thread->func = func;
00656         thread->arg = arg;
00657         thread->result = 0;
00658         return pthread_create(&thread->id, 0, thread_func_wrapper, thread);
00659     #elif defined(JAS_THREADS_WIN32)

```

```

00660         uintptr_t handle;
00661         thread->func = func;
00662         thread->arg = arg;
00663         if (!(handle = _beginthreadex(0, 0, thread_func_wrapper, thread, 0, 0))) {
00664             return -1;
00665         }
00666         thread->id = JAS_CAST(jas_thread_id_t, handle);
00667         return 0;
00668 #endif
00669 }
00670
00671 static inline
00672 int jas_thread_join(jas_thread_t *thread, int *result)
00673 {
00674     assert(thread);
00675 #if defined(JAS_THREADS_C11)
00676     return thrd_join(*thread, result) == thrd_success ? 0 : -1;
00677 #elif defined(JAS_THREADS_PTHREAD)
00678     void *result_buf;
00679     int ret = pthread_join(thread->id, &result_buf);
00680     if (!ret) {
00681         jas_thread_t *other_thread = JAS_CAST(jas_thread_t *, result_buf);
00682         if (result) {
00683             /* A null pointer is probably a bug. */
00684             assert(other_thread);
00685             *result = other_thread ? other_thread->result : 0;
00686         }
00687     }
00688     return ret;
00689 #elif defined(JAS_THREADS_WIN32)
00690     DWORD w;
00691     DWORD code;
00692     if ((w = WaitForSingleObject(thread->id, INFINITE)) != WAIT_OBJECT_0) {
00693         return -1;
00694     }
00695     if (result) {
00696         if (!GetExitCodeThread(thread->id, &code)) {
00697             CloseHandle(thread->id);
00698             return -1;
00699         }
00700         *result = JAS_CAST(int, code);
00701     }
00702     CloseHandle(thread->id);
00703     return 0;
00704 #endif
00705 }
00706
00707 static inline void jas_thread_yield(void)
00708 {
00709 #if defined(JAS_THREADS_C11)
00710     thrd_yield();
00711 #elif defined(JAS_THREADS_PTHREAD)
00712     sched_yield();
00713 #elif defined(JAS_THREADS_WIN32)
00714     SwitchToThread();
00715 #endif
00716 }
00717
00718 #if 0
00719 /* This functionality is not available for all threading support libraries. */
00720 static inline
00721 void jas_thread_exit(int result)
00722 {
00723 #if defined(JAS_THREADS_C11)
00724     thrd_exit(result);
00725 #elif defined(JAS_THREADS_PTHREAD)
00726     /* This does not have a trivial implementation, as far as I can see. */
00727     /* There is no jas_thread_find function. */
00728     jas_thread_t *thread = jas_thread_find(pthread_self());
00729     thread->result = result;
00730     pthread_exit(JAS_CAST(void *, thread));
00731 #endif
00732 }
00733 #endif
00734
00735 static inline
00736 jas_thread_id_t jas_thread_current(void)
00737 {
00738 #if defined(JAS_THREADS_C11)

```

```

00761         return thrd_current();
00762 #elif defined(JAS_THREADS_PTHREAD)
00763         return pthread_self();
00764 #elif defined(JAS_THREADS_WIN32)
00765         /* FIXME - NOT YET IMPLEMENTED. */
00766         abort();
00767 #endif
00768 }
00769 #endif
00770 #endif
00771 #endif
00772
00773 /*****
00774 *
00775 *****/
00776 #endif
00777 #else
00778
00779 /*****
00780 * No Threading Support.
00781 *****/
00782 #endif
00783 #endif
00784 #endif
00785
00790
00791 #ifdef __cplusplus
00792 }
00793 #endif
00794
00795 #endif

```

16.35 jas_tmr.h File Reference

Timer Code.

```

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

```

Functions

- JAS_EXPORT void [jas_tmr_start](#) (jas_tmr_t *tmr)
Start a timer.
- JAS_EXPORT void [jas_tmr_stop](#) (jas_tmr_t *tmr)
Stop a timer.
- JAS_EXPORT double [jas_tmr_get](#) (jas_tmr_t *tmr)
Get the elapsed time for a timer.

16.35.1 Detailed Description

Timer Code.

16.36 jas_tmr.h

[Go to the documentation of this file.](#)

```

00001 /*
00002  * Copyright (c) 2004 Michael David Adams.
00003  * All rights reserved.
00004  */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007  *
00008  * JasPer License Version 2.0
00009  *
00010  * Copyright (c) 2001-2006 Michael David Adams
00011  * Copyright (c) 1999-2000 Image Power, Inc.
00012  * Copyright (c) 1999-2000 The University of British Columbia
00013  *
00014  * All rights reserved.
00015  *
00016  * Permission is hereby granted, free of charge, to any person (the
00017  * "User") obtaining a copy of this software and associated documentation
00018  * files (the "Software"), to deal in the Software without restriction,
00019  * including without limitation the rights to use, copy, modify, merge,
00020  * publish, distribute, and/or sell copies of the Software, and to permit
00021  * persons to whom the Software is furnished to do so, subject to the
00022  * following conditions:
00023  *
00024  * 1. The above copyright notices and this permission notice (which
00025  * includes the disclaimer below) shall be included in all copies or
00026  * substantial portions of the Software.
00027  *
00028  * 2. The name of a copyright holder shall not be used to endorse or
00029  * promote products derived from the Software without specific prior
00030  * written permission.
00031  *
00032  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00054  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058  *
00059  * __END_OF_JASPER_LICENSE__
00060  */
00061
00062
00063 #ifndef JAS_TMR_H
00064 #define JAS_TMR_H
00065
00066 /* The configuration header file should be included first. */
00067 #include <jasper/jas_config.h>
00068
00069 #if defined(JAS_HAVE_SYS_TIME_H)
00070 #include <sys/time.h>
00071 #else
00072 #include <time.h>
00073 #endif
00074
00075 #ifdef __cplusplus

```

```

00080 extern "C" {
00081 #endif
00082
00087
00092 #if defined(JAS_HAVE_GETTIMEOFDAY)
00093
00094 typedef struct {
00095     struct timeval start;
00096     struct timeval stop;
00097 } jas_tmr_t;
00098
00099 #elif defined(JAS_HAVE_GETRUSAGE)
00100
00101 typedef struct {
00102     struct rusage start;
00103     struct rusage stop;
00104 } jas_tmr_t;
00105
00106 #else
00107
00108 typedef int jas_tmr_t;
00109
00110 #endif
00111
00115 JAS_EXPORT
00116 void jas_tmr_start(jas_tmr_t *tmr);
00117
00121 JAS_EXPORT
00122 void jas_tmr_stop(jas_tmr_t *tmr);
00123
00127 JAS_EXPORT
00128 double jas_tmr_get(jas_tmr_t *tmr);
00129
00133
00134 #ifdef __cplusplus
00135 }
00136 #endif
00137
00138 #endif

```

16.37 jas_tvp.h File Reference

Tag/Value Pair Parser.

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

Classes

- struct [jas_taginfo_t](#)
Tag information type.
- struct [jas_tvparker_t](#)
Tag-value parser type.

Functions

- JAS_ATTRIBUTE_PURE JAS_EXPORT const [jas_taginfo_t](#) * [jas_taginfos_lookup](#) (const [jas_taginfo_t](#) *taginfos, const char *name)
Lookup a tag by name.
- JAS_ATTRIBUTE_PURE JAS_EXPORT const [jas_taginfo_t](#) * [jas_taginfo_nonnull](#) (const [jas_taginfo_t](#) *taginfo)
Ensure a nonnull taginfo pointer.

- JAS_EXPORT [jas_tvparser_t](#) * [jas_tvparser_create](#) (const char *s)
Create a tag-value parser for the specified string.
- JAS_EXPORT void [jas_tvparser_destroy](#) ([jas_tvparser_t](#) *tvparser)
Destroy a tag-value parser.
- JAS_EXPORT int [jas_tvparser_next](#) ([jas_tvparser_t](#) *tvparser)
Get the next tag-value pair.
- JAS_ATTRIBUTE_PURE JAS_EXPORT const char * [jas_tvparser_gettag](#) (const [jas_tvparser_t](#) *tvparser)
Get the tag name for the current tag-value pair.
- JAS_ATTRIBUTE_PURE JAS_EXPORT const char * [jas_tvparser_getval](#) (const [jas_tvparser_t](#) *tvparser)
Get the value for the current tag-value pair.

16.37.1 Detailed Description

Tag/Value Pair Parser.

16.38 [jas_tvp.h](#)

[Go to the documentation of this file.](#)

```

00001 /*
00002  * Copyright (c) 2001-2002 Michael David Adams.
00003  * All rights reserved.
00004  */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007  *
00008  * JasPer License Version 2.0
00009  *
00010  * Copyright (c) 2001-2006 Michael David Adams
00011  * Copyright (c) 1999-2000 Image Power, Inc.
00012  * Copyright (c) 1999-2000 The University of British Columbia
00013  *
00014  * All rights reserved.
00015  *
00016  * Permission is hereby granted, free of charge, to any person (the
00017  * "User") obtaining a copy of this software and associated documentation
00018  * files (the "Software"), to deal in the Software without restriction,
00019  * including without limitation the rights to use, copy, modify, merge,
00020  * publish, distribute, and/or sell copies of the Software, and to permit
00021  * persons to whom the Software is furnished to do so, subject to the
00022  * following conditions:
00023  *
00024  * 1. The above copyright notices and this permission notice (which
00025  * includes the disclaimer below) shall be included in all copies or
00026  * substantial portions of the Software.
00027  *
00028  * 2. The name of a copyright holder shall not be used to endorse or
00029  * promote products derived from the Software without specific prior
00030  * written permission.
00031  *
00032  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS

```



```

00046  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00054  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058  *
00059  * __END_OF_JASPER_LICENSE__
00060  */
00061
00062
00063 #ifndef JAS_TVP_H
00064 #define JAS_TVP_H
00065
00066 /*****
00067  * Includes.
00068  \*****/
00069
00070 /* The configuration header file should be included first. */
00071 #include <jasper/jas_config.h>
00072
00073 #ifdef __cplusplus
00074 extern "C" {
00075 #endif
00076
00077 /*****
00078  * Types.
00079  \*****/
00080
00081 typedef struct {
00082     int id;
00083     /* The ID for the tag. */
00084     const char *name;
00085     /* The name of the tag. */
00086 } jas_taginfo_t;
00087
00088 typedef struct {
00089     char *buf;
00090     /* The parsing buffer. */
00091     char *tag;
00092     /* The current tag name. */
00093     const char *val;
00094     /* The current value. */
00095     char *pos;
00096     /* The current position in the parsing buffer. */
00097 } jas_tvparser_t;
00098
00099 /*****
00100  * Tag information functions.
00101  \*****/
00102
00103 JAS_ATTRIBUTE_PURE
00104 JAS_EXPORT
00105 const jas_taginfo_t *jas_taginfos_lookup(const jas_taginfo_t *taginfos,
00106     const char *name);
00107
00108 JAS_ATTRIBUTE_PURE
00109 JAS_EXPORT
00110 const jas_taginfo_t *jas_taginfo_nonnull(const jas_taginfo_t *taginfo);
00111
00112 /*****
00113  * Tag-value parser functions.
00114  \*****/
00115
00116 JAS_EXPORT
00117 jas_tvparser_t *jas_tvparser_create(const char *s);

```

```

00161
00166 JAS_EXPORT
00167 void jas_tvparser_destroy(jas_tvparser_t *tvparser);
00168
00173 JAS_EXPORT
00174 int jas_tvparser_next(jas_tvparser_t *tvparser);
00175
00180 JAS_ATTRIBUTE_PURE
00181 JAS_EXPORT
00182 const char *jas_tvparser_gettag(const jas_tvparser_t *tvparser);
00183
00188 JAS_ATTRIBUTE_PURE
00189 JAS_EXPORT
00190 const char *jas_tvparser_getval(const jas_tvparser_t *tvparser);
00191
00195
00196 #ifdef __cplusplus
00197 }
00198 #endif
00199
00200 #endif

```

16.39 jas_types.h File Reference

Primitive Types.

```

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

```

16.39.1 Detailed Description

Primitive Types.

16.40 jas_types.h

[Go to the documentation of this file.](#)

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2003 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation

```

```

00020 * files (the "Software"), to deal in the Software without restriction,
00021 * including without limitation the rights to use, copy, modify, merge,
00022 * publish, distribute, and/or sell copies of the Software, and to permit
00023 * persons to whom the Software is furnished to do so, subject to the
00024 * following conditions:
00025 *
00026 * 1. The above copyright notices and this permission notice (which
00027 * includes the disclaimer below) shall be included in all copies or
00028 * substantial portions of the Software.
00029 *
00030 * 2. The name of a copyright holder shall not be used to endorse or
00031 * promote products derived from the Software without specific prior
00032 * written permission.
00033 *
00034 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060 *
00061 * __END_OF_JASPER_LICENSE__
00062 */
00063
00068
00069 #ifndef JAS_TYPES_H
00070 #define JAS_TYPES_H
00071
00072 /* The configuration header file should be included first. */
00073 #include <jasper/jas_config.h>
00074
00075 /* Note: The immediately following header files should eventually be removed. */
00076 #include <stddef.h> /* IWYU pragma: export */
00077 #include <stdint.h> /* IWYU pragma: export */
00078
00079 #include <limits.h> /* IWYU pragma: export */
00080
00081 #if defined(JAS_HAVE_SYS_TYPES_H)
00082 #include <sys/types.h> /* IWYU pragma: export */
00083 #endif
00084
00085 #ifdef __cplusplus
00086 extern "C" {
00087 #endif
00088
00089 #define jas_uchar unsigned char
00090 #define jas_uint unsigned int
00091 #define jas_ulong unsigned long
00092 #define jas_longlong long long
00093 #define jas_ulonglong unsigned long long
00094
00095 #if !defined(JAS_NO_SET_SSIZE_T)
00096 #   if !defined(SSIZE_MAX)
00097 #       if (JAS_SIZEOF_INT == JAS_SIZEOF_SIZE_T)
00098 #           define ssize_t int
00099 #           define SSIZE_MAX INT_MAX
00100 #       elif (JAS_SIZEOF_LONG == JAS_SIZEOF_SIZE_T)
00101 #           define ssize_t long
00102 #           define SSIZE_MAX LONG_MAX
00103 #       else
00104 #           define ssize_t jas_longlong

```

```

00105 #                define SSIZE_MAX LLONG_MAX
00106 #                endif
00107 #        endif
00108 #endif
00109
00110 #if 0
00111 #if defined(JAS_HAVE_SSIZE_T)
00112 #define jas_ssize_t ssize_t
00113 #define JAS_SSIZE_MAX SSIZE_MAX
00114 #else
00115 #define jas_ssize_t jas_longlong
00116 #define JAS_SSIZE_MAX LLONG_MAX
00117 #endif
00118 #endif
00119
00120 #if defined(_MSC_VER) && (_MSC_VER < 1800)
00121 #define bool int
00122 #define false 0
00123 #define true 1
00124
00125 #define PRIxFAST32 "x"
00126 #define PRIxFAST16 PRIxFAST32
00127 #define PRIuFAST32 "u"
00128 #define PRIuFAST16 PRIuFAST32
00129 #define PRIiFAST32 "i"
00130 #ifndef _WIN64
00131     #define PRIdPTR "lld"
00132 #else
00133     #define PRIdPTR "d"
00134 #endif
00135
00136 #ifndef _HUGE_ENUF
00137     #define _HUGE_ENUF 1e+300
00138 #endif
00139
00140 #define INFINITY ((float)( _HUGE_ENUF * _HUGE_ENUF))
00141
00142 #define strtoull _strtoui64
00143
00144 #else
00145 #include <stdbool.h> /* IWYU pragma: export */
00146 #include <inttypes.h> /* IWYU pragma: export */
00147 #endif
00148
00149 /* The below macro is intended to be used for type casts. By using this
00150    macro, type casts can be easily located in the source code with
00151    tools like "grep". */
00152 #define JAS_CAST(t, e) \
00153     ((t) (e))
00154
00155 /* The number of bits in the integral type uint_fast32_t. */
00156 /* NOTE: This could underestimate the size on some exotic architectures. */
00157 #define JAS_UINTFAST32_NUMBITS (8 * sizeof(uint_fast32_t))
00158
00159 #if 0
00160 #if defined(JAS_HAVE_MAX_ALIGN_T)
00161 #define jas_max_align_t max_align_t
00162 #else
00163 #define jas_max_align_t long double
00164 #endif
00165 #endif
00166
00167 /*
00168 Assume that a compiler claiming to be compliant with C11 or a later version
00169 of the C standard provides a suitable definition of max_align_t.
00170 The JAS_NO_SET_MAX_ALIGN_T preprocessor symbol can be used to override
00171 this behavior.
00172 */
00173 #if defined(JAS_NO_SET_MAX_ALIGN_T)
00174     /*
00175     The user of this header is assuming responsibility for providing a
00176     suitable definition for max_align_t.
00177     */
00178 #elif defined(_MSC_VER)
00179     /*
00180     Define max_align_t as a preprocessor symbol since using typedef will
00181     cause problems.
00182     */
00183 #    define max_align_t long double
00184 #elif !(defined(__STDC_VERSION__) && (__STDC_VERSION__ - 0 >= 201112L))
00185 #    define max_align_t long double

```

```

00186 #endif
00187
00188 #if 0
00189 #if defined(JAS_HAVE_UINTMAX_T)
00190 #define jas_uintmax_t uintmax_t
00191 #else
00192 #define jas_uintmax_t uint_fast64_t
00193 #endif
00194 #endif
00195
00196 #if 0
00197 #if defined(JAS_HAVE_INTMAX_T)
00198 #define jas_intmax_t intmax_t
00199 #else
00200 #define jas_intmax_t int_fast64_t
00201 #endif
00202 #endif
00203
00204 /* 32-bit unsigned integer type */
00205 typedef uint_least32_t jas_ui32_t;
00206 #define JAS_UI32_MAX UINT_LEAST32_MAX
00207
00208 /* 32-bit signed integer type */
00209 typedef int_least32_t jas_i32_t;
00210 #define JAS_I32_MIN INT_LEAST32_MIN
00211 #define JAS_I32_MAX INT_LEAST32_MAX
00212
00213 #ifdef __cplusplus
00214 }
00215 #endif
00216
00217 #endif

```

16.41 jas_version.h File Reference

JasPer Version.

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

16.41.1 Detailed Description

JasPer Version.

16.42 jas_version.h

[Go to the documentation of this file.](#)

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2003 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *

```

```

00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00068
00069 #ifndef JAS_VERSION_H
00070 #define JAS_VERSION_H
00071
00072 /* The configuration header file should be included first. */
00073 #include <jasper/jas_config.h>
00074
00075 #ifdef __cplusplus
00076 extern "C" {
00077 #endif
00078
00079 /*****
00080  * Constants and types.
00081  \*****/
00082
00083 #define JAS_COPYRIGHT \
00084     "Copyright (c) 2001-2022 Michael David Adams.\n" \
00085     "Copyright (c) 1999-2000 Image Power, Inc. and the University of\n" \
00086     "  British Columbia.\n" \
00087     "All rights reserved.\n"
00088
00089 #define JAS_NOTES \
00090     "For information about the JasPer project, see:\n" \
00091     "  https://jasper-software.github.io/jasper\n" \
00092     "  https://www.ece.uvic.ca/~mdadams/jasper\n" \
00093     "For online documentation on the JasPer software, see:\n" \
00094     "  https://jasper-software.github.io/jasper-manual\n" \
00095     "Please submit bug reports using the bug-tracker at:\n" \
00096     "  https://github.com/jasper-software/jasper/issues\n"
00097
00098 /*****
00099  * Functions.
00100  \*****/

```

```

00101
00102 JAS_ATTRIBUTE_CONST
00103 JAS_EXPORT
00104 const char *jas_getversion(void);
00105 /* Get the version information for the Jasper library. */
00106 /* Note: Since libjasper can be built as a shared library, the version
00107    returned by this function may not necessarily correspond to JAS_VERSION. */
00108
00109 #ifdef __cplusplus
00110 }
00111 #endif
00112
00113 #endif

```

16.43 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>
#include <jasper/jas_thread.h>
#include <jasper/jas_log.h>

```

16.43.1 Detailed Description

JasPer Main Header.

16.44 jasper.h

[Go to the documentation of this file.](#)

```

00001 /*
00002  * Copyright (c) 2001-2003 Michael David Adams.
00003  * All rights reserved.
00004  */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007  *

```

```

00008  * JasPer License Version 2.0
00009  *
00010  * Copyright (c) 2001-2006 Michael David Adams
00011  * Copyright (c) 1999-2000 Image Power, Inc.
00012  * Copyright (c) 1999-2000 The University of British Columbia
00013  *
00014  * All rights reserved.
00015  *
00016  * Permission is hereby granted, free of charge, to any person (the
00017  * "User") obtaining a copy of this software and associated documentation
00018  * files (the "Software"), to deal in the Software without restriction,
00019  * including without limitation the rights to use, copy, modify, merge,
00020  * publish, distribute, and/or sell copies of the Software, and to permit
00021  * persons to whom the Software is furnished to do so, subject to the
00022  * following conditions:
00023  *
00024  * 1. The above copyright notices and this permission notice (which
00025  * includes the disclaimer below) shall be included in all copies or
00026  * substantial portions of the Software.
00027  *
00028  * 2. The name of a copyright holder shall not be used to endorse or
00029  * promote products derived from the Software without specific prior
00030  * written permission.
00031  *
00032  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037  * PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00054  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058  *
00059  * __END_OF_JASPER_LICENSE__
00060  */
00061
00066
00067 #ifndef JAS_JASPER_H
00068 #define JAS_JASPER_H
00069
00070 // IWYU pragma: begin_exports
00071
00072 /* The configuration header file should be included first. */
00073 #include <jasper/jas_config.h>
00074
00075 #include <jasper/jas_types.h>
00076 #include <jasper/jas_version.h>
00077
00078 #include <jasper/jas_init.h>
00079 #include <jasper/jas_cm.h>
00080 #include <jasper/jas_icc.h>
00081 #include <jasper/jas_fix.h>
00082 #include <jasper/jas_debug.h>
00083 #include <jasper/jas_getopt.h>
00084 #include <jasper/jas_image.h>
00085 #include <jasper/jas_math.h>
00086 #include <jasper/jas_malloc.h>
00087 #include <jasper/jas_seq.h>
00088 #include <jasper/jas_stream.h>
00089 #include <jasper/jas_string.h>
00090 #include <jasper/jas_tmr.h>
00091 #include <jasper/jas_tvp.h>
00092 #include <jasper/jas_thread.h>

```



```

00093 #include <jasper/jas_init.h>
00094 #include <jasper/jas_log.h>
00095
00096 #ifdef __cplusplus
00097 extern "C" {
00098 #endif
00099
00100 #ifdef __cplusplus
00101 }
00102 #endif
00103
00104 // IWYU pragma: end_exports
00105
00106 #endif

```

16.45 jp2_cod.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.

```

```

00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00064  /*
00065  * JP2 Library
00066  *
00067  * $Id$
00068  */
00069
00070 #ifndef JP2_COD_H
00071 #define JP2_COD_H
00072
00073 /*****
00074  * Includes.
00075  *****/
00076
00077 #include "jasper/jas_types.h"
00078 #include "jasper/jas_stream.h"
00079 #include "jasper/jas_image.h"
00080
00081 #include <stdio.h>
00082
00083 /*****
00084  * Macros.
00085  *****/
00086
00087 #define JP2_SPTOBPC(s, p) \
00088     (((p) - 1) & 0x7f) | (((s) & 1) << 7))
00089
00090 /*****
00091  * Box class.
00092  *****/
00093
00094 #define JP2_BOX_HDRLEN(ext) ((ext) ? 16 : 8)
00095
00096 /* Box types. */
00097 #define JP2_BOX_JP          0x6a502020 /* Signature */
00098 #define JP2_BOX_FTYPE      0x66747970 /* File Type */
00099 #define JP2_BOX_JP2H       0x6a703268 /* JP2 Header */
00100 #define JP2_BOX_IHDR       0x69686472 /* Image Header */
00101 #define JP2_BOX_BPCC       0x62706363 /* Bits Per Component */
00102 #define JP2_BOX_COLR       0x63666c72 /* Color Specification */
00103 #define JP2_BOX_PCLR       0x70636c72 /* Palette */
00104 #define JP2_BOX_CMAP       0x636d6170 /* Component Mapping */
00105 #define JP2_BOX_CDEF       0x63646566 /* Channel Definition */
00106 #define JP2_BOX_RES        0x72657320 /* Resolution */
00107 #define JP2_BOX_RESC       0x72657363 /* Capture Resolution */
00108 #define JP2_BOX_RESD       0x72657364 /* Default Display Resolution */
00109 #define JP2_BOX_JP2C       0x6a703263 /* Contiguous Code Stream */
00110 #define JP2_BOX_JP2I       0x6a703269 /* Intellectual Property */
00111 #define JP2_BOX_XML        0x786d6c20 /* XML */
00112 #define JP2_BOX_UUID       0x75756964 /* UUID */
00113 #define JP2_BOX_UINF       0x75696e66 /* UUID Info */
00114 #define JP2_BOX_ULST       0x75637374 /* UUID List */
00115 #define JP2_BOX_URL        0x75726c20 /* URL */
00116
00117 #define JP2_BOX_SUPER      0x01
00118 #define JP2_BOX_NODATA     0x02
00119
00120 /* JP box data. */
00121
00122 #define JP2_JP_MAGIC       0x0d0a870a
00123 #define JP2_JP_LEN        12
00124
00125 typedef struct {
00126     uint_fast32_t magic;
00127 } jp2_jp_t;
00128
00129 /* FTYPE box data. */
00130
00131 #define JP2_FTYPE_MAXCOMPATCODES 32
00132 #define JP2_FTYPE_MAJVER      0x6a703220
00133 #define JP2_FTYPE_MINVER      0
00134 #define JP2_FTYPE_COMPATCODE   JP2_FTYPE_MAJVER
00135
00136 typedef struct {
00137     uint_fast32_t majver;
00138     uint_fast32_t minver;
00139     uint_fast32_t numcompatcodes;
00140     uint_fast32_t compatcodes[JP2_FTYPE_MAXCOMPATCODES];

```

```

00141 } jp2_ftyp_t;
00142
00143 /* IHDR box data. */
00144
00145 #define JP2_IHDR_COMPTYPE      7
00146 #define JP2_IHDR_BPCNULL      255
00147
00148 typedef struct {
00149     uint_fast32_t width;
00150     uint_fast32_t height;
00151     uint_fast16_t numcmpts;
00152     uint_fast8_t bpc;
00153     uint_fast8_t comptype;
00154     uint_fast8_t csunk;
00155     uint_fast8_t ipr;
00156 } jp2_ihdr_t;
00157
00158 /* BPCC box data. */
00159
00160 typedef struct {
00161     uint_fast16_t numcmpts;
00162     uint_fast8_t *bpcs;
00163 } jp2_bpcc_t;
00164
00165 /* COLR box data. */
00166
00167 #define JP2_COLR_ENUM      1
00168 #define JP2_COLR_ICC      2
00169 #define JP2_COLR_PRI      0
00170
00171 #define JP2_COLR_SRGB      16
00172 #define JP2_COLR_SGRAY      17
00173 #define JP2_COLR_SYCC      18
00174
00175 typedef struct {
00176     uint_fast8_t method;
00177     uint_fast8_t pri;
00178     uint_fast8_t approx;
00179     uint_fast32_t csid;
00180     jas_uchar *iccp;
00181     size_t iccplen;
00182     /* XXX - Someday we ought to add ICC profile data here. */
00183 } jp2_colr_t;
00184
00185 /* PCLR box data. */
00186
00187 typedef struct {
00188     uint_fast16_t numlutents;
00189     uint_fast8_t numchans;
00190     int_fast32_t *lutdata;
00191     uint_fast8_t *bpc;
00192 } jp2_pclr_t;
00193
00194 /* CDEF box per-channel data. */
00195
00196 #define JP2_CDEF_RGB_R      1
00197 #define JP2_CDEF_RGB_G      2
00198 #define JP2_CDEF_RGB_B      3
00199
00200 #define JP2_CDEF_YCBCR_Y      1
00201 #define JP2_CDEF_YCBCR_CB      2
00202 #define JP2_CDEF_YCBCR_CR      3
00203
00204 #define JP2_CDEF_GRAY_Y      1
00205
00206 #define JP2_CDEF_TYPE_COLOR      0
00207 #define JP2_CDEF_TYPE_OPACITY      1
00208 #define JP2_CDEF_TYPE_UNSPEC      65535
00209 #define JP2_CDEF_ASOC_ALL      0
00210 #define JP2_CDEF_ASOC_NONE      65535
00211
00212 typedef struct {
00213     uint_fast16_t channo;
00214     uint_fast16_t type;
00215     uint_fast16_t assoc;
00216 } jp2_cdefchan_t;
00217
00218 /* CDEF box data. */
00219
00220 typedef struct {
00221     uint_fast16_t numchans;

```

```

00222     jp2_cdefchan_t *ents;
00223 } jp2_cdef_t;
00224
00225 typedef struct {
00226     uint_fast16_t cmptno;
00227     uint_fast8_t map;
00228     uint_fast8_t pcol;
00229 } jp2_cmapent_t;
00230
00231 typedef struct {
00232     uint_fast16_t numchans;
00233     jp2_cmapent_t *ents;
00234 } jp2_cmap_t;
00235
00236 #define JP2_CMAP_DIRECT      0
00237 #define JP2_CMAP_PALETTE    1
00238
00239 /* Generic box. */
00240
00241 struct jp2_boxops_s;
00242 typedef struct {
00243
00244     const struct jp2_boxops_s *ops;
00245     const struct jp2_boxinfo_s *info;
00246
00247     uint_fast32_t type;
00248
00249     /* The length of the box including the (variable-length) header. */
00250     uint_fast32_t len;
00251
00252     /* The length of the box data. */
00253     uint_fast32_t datalen;
00254
00255     union {
00256         jp2_jp_t jp;
00257         jp2_ftyp_t ftyp;
00258         jp2_ihdr_t ihdr;
00259         jp2_bpcc_t bpcc;
00260         jp2_colr_t colr;
00261         jp2_pclr_t pclr;
00262         jp2_cdef_t cdef;
00263         jp2_cmap_t cmap;
00264     } data;
00265 } jp2_box_t;
00266
00267 typedef struct jp2_boxops_s {
00268     void (*init)(jp2_box_t *box);
00269     void (*destroy)(jp2_box_t *box);
00270     int (*getdata)(jp2_box_t *box, jas_stream_t *in);
00271     int (*putdata)(const jp2_box_t *box, jas_stream_t *out);
00272     void (*dumpdata)(const jp2_box_t *box);
00273 } jp2_boxops_t;
00274
00275 /******\
00276 *
00277 \*****/
00278
00279 typedef struct jp2_boxinfo_s {
00280     int type;
00281     int flags;
00282     const char *name;
00283     jp2_boxops_t ops;
00284 } jp2_boxinfo_t;
00285
00286 /******\
00287 * Box class.
00288 \*****/
00289
00290 jp2_box_t *jp2_box_create(int type);
00291 void jp2_box_destroy(jp2_box_t *box);
00292 jp2_box_t *jp2_box_get(jas_stream_t *in);
00293 int jp2_box_put(jp2_box_t *box, jas_stream_t *out);
00294
00295 JAS_ATTRIBUTE_CONST
00296 static inline uint_least8_t JP2_DTYPETOBC(uint_least8_t dtype)
00297 {
00298     return (JAS_IMAGE_CDT_GETSGND(dtype) << 7) | (JAS_IMAGE_CDT_GETPREC(dtype) - 1);
00299 }
00300
00301 JAS_ATTRIBUTE_CONST

```

```

00303 static inline uint_least8_t JP2_BPCTODTYPE(uint_least8_t bpc)
00304 {
00305     return JAS_IMAGE_CDT_SETSGND(bpc » 7) | JAS_IMAGE_CDT_SETPREC((bpc & 0x7f) + 1);
00306 }
00307
00308 #define ICC_CS_RGB      0x52474220
00309 #define ICC_CS_YCBCR    0x59436272
00310 #define ICC_CS_GRAY     0x47524159
00311
00312 const jp2_cdefchan_t *jp2_cdef_lookup(jp2_cdef_t *cdef, int channo);
00313
00314 #endif

```

16.46 jp2_dec.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__

```

```

00062 */
00063
00064 #ifndef JP2_DEC_H
00065 #define JP2_DEC_H
00066
00067 #include "jasper/jas_image.h"
00068 #include "jp2_cod.h"
00069
00070 typedef struct {
00071
00072     jp2_box_t *pclr;
00073     jp2_box_t *cdef;
00074     jp2_box_t *ihdr;
00075     jp2_box_t *bpcc;
00076     jp2_box_t *cmap;
00077     jp2_box_t *colr;
00078     jas_image_t *image;
00079     uint_fast16_t numchans;
00080     uint_fast16_t *chantocmptlut;
00081
00082 } jp2_dec_t;
00083
00084 #endif

```

16.47 jpc_bs.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE

```

```

00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY.  THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES").  THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00064  /*
00065  * Bit Stream Class
00066  *
00067  * $Id$
00068  */
00069
00070 #ifndef JPC_BS_H
00071 #define JPC_BS_H
00072
00073 /*****
00074  * Includes.
00075  *****/
00076
00077 #include "jasper/jas_types.h"
00078 #include "jasper/jas_stream.h"
00079
00080 #include <assert.h>
00081 #include <stdio.h>
00082
00083 /*****
00084  * Constants.
00085  *****/
00086
00087 /*
00088  * Bit stream open mode flags.
00089  */
00090
00091 /* Bit stream open for reading. */
00092 #define JPC_BITSTREAM_READ 0x01
00093 /* Bit stream open for writing. */
00094 #define JPC_BITSTREAM_WRITE 0x02
00095
00096 /*
00097  * Bit stream flags.
00098  */
00099
00100 /* Do not close underlying character stream. */
00101 #define JPC_BITSTREAM_NOCLOSE 0x01
00102 /* End of file has been reached while reading. */
00103 #define JPC_BITSTREAM_EOF 0x02
00104 /* An I/O error has occurred. */
00105 #define JPC_BITSTREAM_ERR 0x04
00106
00107 /*****
00108  * Types.
00109  *****/
00110
00111 /* Bit stream class. */
00112
00113 typedef struct {
00114
00115     /* Some miscellaneous flags. */
00116     int flags_;
00117
00118     /* The input/output buffer. */
00119     uint_fast16_t buf_;
00120
00121     /* The number of bits remaining in the byte being read/written. */
00122     int cnt_;
00123
00124     /* The underlying stream associated with this bit stream. */
00125     jas_stream_t *stream_;
00126
00127     /* The mode in which this bit stream was opened. */
00128     int openmode_;
00129
00130 } jpc_bitstream_t;
00131

```

```

00132 /*****
00133 * Functions/macros for opening and closing bit streams..
00134 \*****/
00135
00136 /* Open a stream as a bit stream. */
00137 jpc_bitstream_t *jpc_bitstream_sopen(jas_stream_t *stream, const char *mode);
00138
00139 /* Close a bit stream. */
00140 int jpc_bitstream_close(jpc_bitstream_t *bitstream);
00141
00142 /*****
00143 * Functions/macros for reading from and writing to bit streams..
00144 \*****/
00145
00146 /* Read a bit from a bit stream. */
00147 #ifndef NDEBUG
00148 #define jpc_bitstream_getbit(bitstream) \
00149     jpc_bitstream_getbit_func(bitstream)
00150 #else
00151 #define jpc_bitstream_getbit(bitstream) \
00152     jpc_bitstream_getbit_macro(bitstream)
00153 #endif
00154
00155 /* Write a bit to a bit stream. */
00156 #ifndef NDEBUG
00157 #define jpc_bitstream_putbit(bitstream, v) \
00158     jpc_bitstream_putbit_func(bitstream, v)
00159 #else
00160 #define jpc_bitstream_putbit(bitstream, v) \
00161     jpc_bitstream_putbit_macro(bitstream, v)
00162 #endif
00163
00164 /* Read one or more bits from a bit stream. */
00165 long jpc_bitstream_getbits(jpc_bitstream_t *bitstream, int n);
00166
00167 /* Write one or more bits to a bit stream. */
00168 int jpc_bitstream_putbits(jpc_bitstream_t *bitstream, int n, long v);
00169
00170 /*****
00171 * Functions/macros for flushing and aligning bit streams.
00172 \*****/
00173
00174 /* Align the current position within the bit stream to the next byte
00175    boundary. */
00176 int jpc_bitstream_align(jpc_bitstream_t *bitstream);
00177
00178 /* Align the current position in the bit stream with the next byte boundary,
00179    ensuring that certain bits consumed in the process match a particular
00180    pattern. */
00181 int jpc_bitstream_inalign(jpc_bitstream_t *bitstream, int fillmask,
00182    int filldata);
00183
00184 /* Align the current position in the bit stream with the next byte boundary,
00185    writing bits from the specified pattern (if necessary) in the process. */
00186 int jpc_bitstream_outalign(jpc_bitstream_t *bitstream, int filldata);
00187
00188 /* Check if a bit stream needs alignment. */
00189 JAS_ATTRIBUTE_PURE
00190 int jpc_bitstream_needsalign(const jpc_bitstream_t *bitstream);
00191
00192 /* How many additional bytes would be output if the bit stream was aligned? */
00193 JAS_ATTRIBUTE_PURE
00194 int jpc_bitstream_pending(const jpc_bitstream_t *bitstream);
00195
00196 /*****
00197 * Functions/macros for querying state information for bit streams.
00198 \*****/
00199
00200 /* Has EOF been encountered on a bit stream? */
00201 #define jpc_bitstream_eof(bitstream) \
00202     ((bitstream)->flags_ & JPC_BITSTREAM_EOF)
00203
00204 /*****
00205 * Internals.
00206 \*****/
00207
00208 /* DO NOT DIRECTLY INVOKE ANY OF THE MACROS OR FUNCTIONS BELOW. THEY ARE
00209    FOR INTERNAL USE ONLY. */
00210
00211 int jpc_bitstream_getbit_func(jpc_bitstream_t *bitstream);
00212

```



```

00213 int jpc_bitstream_putbit_func(jpc_bitstream_t *bitstream, int v);
00214
00215 int jpc_bitstream_fillbuf(jpc_bitstream_t *bitstream);
00216
00217 #define jpc_bitstream_getbit_macro(bitstream) \
00218     (assert((bitstream)->openmode_ & JPC_BITSTREAM_READ), \
00219      (--(bitstream)->cnt_ >= 0) ? \
00220      ((int)((bitstream)->buf_ » (bitstream)->cnt_) & 1)) : \
00221      jpc_bitstream_fillbuf(bitstream))
00222
00223 #define jpc_bitstream_putbit_macro(bitstream, bit) \
00224     (assert((bitstream)->openmode_ & JPC_BITSTREAM_WRITE), \
00225      (--(bitstream)->cnt_ < 0) ? \
00226      ((bitstream)->buf_ = ((bitstream)->buf_ « 8) & 0xffff, \
00227      (bitstream)->cnt_ = ((bitstream)->buf_ == 0xff00) ? 6 : 7, \
00228      (bitstream)->buf_ |= ((bit) & 1) « (bitstream)->cnt_, \
00229      (jas_stream_putc((bitstream)->stream_, (bitstream)->buf_ » 8) == EOF) \
00230      ? (EOF) : ((bit) & 1)) : \
00231      ((bitstream)->buf_ |= ((bit) & 1) « (bitstream)->cnt_, \
00232      (bit) & 1))
00233
00234 #endif

```

16.48 jpc_cod.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE

```

```

00052 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060 *
00061 * __END_OF_JASPER_LICENSE__
00062 */
00063
00064 /*
00065 * $Id$
00066 */
00067
00068 #ifndef JPC_COD_H
00069 #define JPC_COD_H
00070
00071 #include "jpc_tlcod.h"
00072
00073 /*****
00074 * Constants.
00075 *****/
00076
00077 /* The nominal word size used by this implementation. */
00078 #define JPC_PREC 32
00079
00080 void jpc_init(void);
00081
00082 #endif

```

16.49 jpc_cs.h

```

00001 /*
00002 * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003 * British Columbia.
00004 * Copyright (c) 2001-2002 Michael David Adams.
00005 * All rights reserved.
00006 */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009 *
00010 * JasPer License Version 2.0
00011 *
00012 * Copyright (c) 2001-2006 Michael David Adams
00013 * Copyright (c) 1999-2000 Image Power, Inc.
00014 * Copyright (c) 1999-2000 The University of British Columbia
00015 *
00016 * All rights reserved.
00017 *
00018 * Permission is hereby granted, free of charge, to any person (the
00019 * "User") obtaining a copy of this software and associated documentation
00020 * files (the "Software"), to deal in the Software without restriction,
00021 * including without limitation the rights to use, copy, modify, merge,
00022 * publish, distribute, and/or sell copies of the Software, and to permit
00023 * persons to whom the Software is furnished to do so, subject to the
00024 * following conditions:
00025 *
00026 * 1. The above copyright notices and this permission notice (which
00027 * includes the disclaimer below) shall be included in all copies or
00028 * substantial portions of the Software.
00029 *
00030 * 2. The name of a copyright holder shall not be used to endorse or
00031 * promote products derived from the Software without specific prior
00032 * written permission.
00033 *
00034 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,

```

```

00043 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060 *
00061 * __END_OF_JASPER_LICENSE__
00062 */
00063
00064 /*
00065 * JPEG-2000 Code Stream Library
00066 *
00067 * $Id$
00068 */
00069
00070 #ifndef JPC_CS_H
00071 #define JPC_CS_H
00072
00073 /*****\
00074 * Includes.
00075 \*****/
00076
00077 #include "jasper/jas_stream.h"
00078
00079 #include <assert.h>
00080 #include <stdio.h>
00081
00082 /*****\
00083 * Constants and Types.
00084 \*****/
00085
00086 /* The maximum number of resolution levels. */
00087 #define JPC_MAXRLVLS 33
00088
00089 /* The maximum number of bands. */
00090 #define JPC_MAXBANDS (3 * JPC_MAXRLVLS + 1)
00091
00092 /* The maximum number of layers. */
00093 #define JPC_MAXLYRS 16384
00094
00095 /*****\
00096 * Code stream.
00097 \*****/
00098
00099 /*
00100 * Code stream states.
00101 */
00102
00103 /* Initial. */
00104 #define JPC_CS_INIT 0
00105 /* Main header. */
00106 #define JPC_CS_MHDR 1
00107 /* Tile-part header. */
00108 #define JPC_CS_THDR 2
00109 /* Main trailer. */
00110 #define JPC_CS_MTLR 3
00111 /* Tile-part data. */
00112 #define JPC_CS_TDATA 4
00113
00114 /*
00115 * Unfortunately, the code stream syntax was not designed in such a way that
00116 * any given marker segment can be correctly decoded without additional state
00117 * derived from previously decoded marker segments.
00118 * For example, a RGN/COC/QCC marker segment cannot be decoded unless the
00119 * number of components is known.
00120 */
00121
00122 /*
00123 * Code stream state information.

```

```

00124  */
00125
00126 typedef struct {
00127
00128     /* The number of components. */
00129     uint_fast16_t numcomps;
00130
00131 } jpc_cstate_t;
00132
00133 /*****\
00134  * SOT marker segment parameters.
00135  \*****/
00136
00137 typedef struct {
00138
00139     /* The tile number. */
00140     uint_fast16_t tileno;
00141
00142     /* The combined length of the marker segment and its auxiliary data
00143        (i.e., packet data). */
00144     uint_fast32_t len;
00145
00146     /* The tile-part instance. */
00147     uint_fast8_t partno;
00148
00149     /* The number of tile-parts. */
00150     uint_fast8_t numparts;
00151
00152 } jpc_sot_t;
00153
00154 /*****\
00155  * SI2 marker segment parameters.
00156  \*****/
00157
00158 /* Per component information. */
00159
00160 typedef struct {
00161
00162     /* The precision of the samples. */
00163     uint_fast8_t prec;
00164
00165     /* The signedness of the samples. */
00166     uint_fast8_t sgnd;
00167
00168     /* The horizontal separation of samples with respect to the reference
00169        grid. */
00170     uint_fast8_t hsamp;
00171
00172     /* The vertical separation of samples with respect to the reference
00173        grid. */
00174     uint_fast8_t vsamp;
00175
00176 } jpc_sizcomp_t;
00177
00178 /* SI2 marker segment parameters. */
00179
00180 typedef struct {
00181
00182     /* The code stream capabilities. */
00183     uint_fast16_t caps;
00184
00185     /* The width of the image in units of the reference grid. */
00186     uint_fast32_t width;
00187
00188     /* The height of the image in units of the reference grid. */
00189     uint_fast32_t height;
00190
00191     /* The horizontal offset from the origin of the reference grid to the
00192        left side of the image area. */
00193     uint_fast32_t xoff;
00194
00195     /* The vertical offset from the origin of the reference grid to the
00196        top side of the image area. */
00197     uint_fast32_t yoff;
00198
00199     /* The nominal width of a tile in units of the reference grid. */
00200     uint_fast32_t tilewidth;
00201
00202     /* The nominal height of a tile in units of the reference grid. */
00203     uint_fast32_t tileheight;
00204

```

```

00205      /* The horizontal offset from the origin of the reference grid to the
00206         left side of the first tile. */
00207      uint_fast32_t tilexoff;
00208
00209      /* The vertical offset from the origin of the reference grid to the
00210         top side of the first tile. */
00211      uint_fast32_t tileyoff;
00212
00213      /* The number of components. */
00214      uint_fast16_t numcomps;
00215
00216      /* The per-component information. */
00217      jpc_sizcomp_t *comps;
00218
00219 } jpc_siz_t;
00220
00221 /*****\
00222  * COD marker segment parameters.
00223  \*****/
00224
00225 /*
00226  * Coding style constants.
00227  */
00228
00229 /* Precincts may be used. */
00230 #define JPC_COX_PRT      0x01
00231 /* SOP marker segments may be used. */
00232 #define JPC_COD_SOP      0x02
00233 /* EPH marker segments may be used. */
00234 #define JPC_COD_EPH      0x04
00235
00236 /*
00237  * Progression order constants.
00238  */
00239
00240 /* Layer-resolution-component-precinct progressive
00241    (i.e., progressive by fidelity). */
00242 #define JPC_COD_LRCPPRG  0
00243 /* Resolution-layer-component-precinct progressive
00244    (i.e., progressive by resolution). */
00245 #define JPC_COD_RLCPPRG  1
00246 /* Resolution-precinct-component-layer progressive. */
00247 #define JPC_COD_RPCLPRG  2
00248 /* Precinct-component-resolution-layer progressive. */
00249 #define JPC_COD_PCRLPRG  3
00250 /* Component-position-resolution-layer progressive. */
00251 #define JPC_COD_CPRLPRG  4
00252
00253 /*
00254  * Code block style constants.
00255  */
00256
00257 #define JPC_COX_LAZY      0x01 /* Selective arithmetic coding bypass. */
00258 #define JPC_COX_RESET     0x02 /* Reset context probabilities. */
00259 #define JPC_COX_TERMALL   0x04 /* Terminate all coding passes. */
00260 #define JPC_COX_VSC       0x08 /* Vertical stripe causal context formation. */
00261 #define JPC_COX_PTERM     0x10 /* Predictable termination. */
00262 #define JPC_COX_SEGSYM    0x20 /* Use segmentation symbols. */
00263
00264 /* Transform constants. */
00265 #define JPC_COX_INS       0x00 /* Irreversible 9/7. */
00266 #define JPC_COX_RFT       0x01 /* Reversible 5/3. */
00267
00268 /* Multicomponent transform constants. */
00269 #define JPC_COD_NOMCT     0x00 /* No multicomponent transform. */
00270 #define JPC_COD_MCT       0x01 /* Multicomponent transform. */
00271
00272 /* Get the code block size value from the code block size exponent. */
00273 JAS_ATTRIBUTE_CONST
00274 static inline unsigned JPC_COX_CBLKSIZEEXPN(unsigned x)
00275 {
00276     return x - 2;
00277 }
00278
00279 /* Get the code block size exponent from the code block size value. */
00280 JAS_ATTRIBUTE_CONST
00281 static inline unsigned JPC_COX_GETCBLKSIZEEXPN(unsigned x)
00282 {
00283     return x + 2;
00284 }
00285

```

```

00286 /* Per resolution-level information. */
00287
00288 typedef struct {
00289
00290     /* The packet partition width. */
00291     uint_fast8_t parwidthval;
00292
00293     /* The packet partition height. */
00294     uint_fast8_t parheightval;
00295
00296 } jpc_coxrlvl_t;
00297
00298 /* Per component information. */
00299
00300 typedef struct {
00301
00302     /* The coding style. */
00303     uint_fast8_t csty;
00304
00305     /* The number of decomposition levels. */
00306     uint_fast8_t numdlvls;
00307
00308     /* The nominal code block width specifier. */
00309     uint_fast8_t cblkwidthval;
00310
00311     /* The nominal code block height specifier. */
00312     uint_fast8_t cblkheightval;
00313
00314     /* The style of coding passes. */
00315     uint_fast8_t cblksty;
00316
00317     /* The QMFB employed. */
00318     uint_fast8_t qmfbid;
00319
00320     /* The number of resolution levels. */
00321     int numrlvls;
00322
00323     /* The per-resolution-level information. */
00324     jpc_coxrlvl_t rlvls[JPC_MAXRLVLS];
00325
00326 } jpc_coxcp_t;
00327
00328 /* COD marker segment parameters. */
00329
00330 typedef struct {
00331
00332     /* The general coding style. */
00333     uint_fast8_t csty;
00334
00335     /* The progression order. */
00336     uint_fast8_t prg;
00337
00338     /* The number of layers. */
00339     uint_fast16_t numlyrs;
00340
00341     /* The multicomponent transform. */
00342     uint_fast8_t mctrans;
00343
00344     /* Component-related parameters. */
00345     jpc_coxcp_t compparms;
00346
00347 } jpc_cod_t;
00348
00349 /* COC marker segment parameters. */
00350
00351 typedef struct {
00352
00353     /* The component number. */
00354     uint_fast16_t compno;
00355
00356     /* Component-related parameters. */
00357     jpc_coxcp_t compparms;
00358
00359 } jpc_coc_t;
00360
00361 /*****\
00362 * RGN marker segment parameters.
00363 \*****/
00364
00365 /* The maxshift ROI style. */
00366 #define JPC_RGN_MAXSHIFT 0x00

```

```

00367
00368 typedef struct {
00369
00370     /* The component to which the marker applies. */
00371     uint_fast16_t compno;
00372
00373     /* The ROI style. */
00374     uint_fast8_t roisty;
00375
00376     /* The ROI shift value. */
00377     uint_fast8_t roishift;
00378
00379 } jpc_rgn_t;
00380
00381 /******\
00382  * QCD/QCC marker segment parameters.
00383  \*****/
00384
00385 /*
00386  * Quantization style constants.
00387  */
00388
00389 #define JPC_QCX_NOQNT    0 /* No quantization. */
00390 #define JPC_QCX_SQNT    1 /* Scalar quantization, implicit. */
00391 #define JPC_QCX_SEQNT    2 /* Scalar quantization, explicit. */
00392
00393 /*
00394  * Stepsize manipulation macros.
00395  */
00396
00397 JAS_ATTRIBUTE_CONST
00398 static inline unsigned JPC_QCX_GETEXPN(unsigned x)
00399 {
00400     return x » 11;
00401 }
00402
00403 JAS_ATTRIBUTE_CONST
00404 static inline unsigned JPC_QCX_GETMANT(unsigned x)
00405 {
00406     return x & 0x7ff;
00407 }
00408
00409 JAS_ATTRIBUTE_CONST
00410 static inline uint_fast16_t JPC_QCX_EXPX(unsigned x)
00411 {
00412     assert(!(x & (~0x1f)));
00413
00414     return (x & 0x1f) « 11;
00415 }
00416
00417 JAS_ATTRIBUTE_CONST
00418 static inline uint_fast16_t JPC_QCX_MANT(unsigned x)
00419 {
00420     assert(!(x & (~0x7ff)));
00421
00422     return x & 0x7ff;
00423 }
00424
00425 /* Per component information. */
00426
00427 typedef struct {
00428
00429     /* The quantization style. */
00430     uint_fast8_t qntsty;
00431
00432     /* The number of step sizes. */
00433     int numstepsizes;
00434
00435     /* The step sizes. */
00436     uint_fast16_t *stepsizes;
00437
00438     /* The number of guard bits. */
00439     uint_fast8_t numguard;
00440
00441 } jpc_qxcpc_t;
00442
00443 /* QCC marker segment parameters. */
00444
00445 typedef struct {
00446
00447     /* The component associated with this marker segment. */

```

```

00448         uint_fast16_t compno;
00449
00450         /* The parameters. */
00451         jpc_qcxcpc_t compparms;
00452
00453     } jpc_qcc_t;
00454
00455     /* QCD marker segment parameters. */
00456
00457     typedef struct {
00458
00459         /* The parameters. */
00460         jpc_qcxcpc_t compparms;
00461
00462     } jpc_qcd_t;
00463
00464     /******\
00465     * POD marker segment parameters.
00466     \*****/
00467
00468     typedef struct {
00469
00470         /* The progression order. */
00471         uint_fast8_t prgord;
00472
00473         /* The lower bound (inclusive) on the resolution level for the
00474         progression order volume. */
00475         uint_fast8_t rlvlnostart;
00476
00477         /* The upper bound (exclusive) on the resolution level for the
00478         progression order volume. */
00479         uint_fast8_t rlvlnoend;
00480
00481         /* The lower bound (inclusive) on the component for the progression
00482         order volume. */
00483         uint_fast16_t compnoend;
00484
00485         /* The upper bound (exclusive) on the component for the progression
00486         order volume. */
00487         uint_fast16_t compnoend;
00488
00489         /* The upper bound (exclusive) on the layer for the progression
00490         order volume. */
00491         uint_fast16_t lyrnoend;
00492
00493     } jpc_pocpchgt_t;
00494
00495     /* An alias for the above type. */
00496     typedef jpc_pocpchgt_t jpc_pchg_t;
00497
00498     /* POC marker segment parameters. */
00499
00500     typedef struct {
00501
00502         /* The number of progression order changes. */
00503         int numpchgs;
00504
00505         /* The per-progression-order-change information. */
00506         jpc_pocpchgt_t *pchgs;
00507
00508     } jpc_poc_t;
00509
00510     /******\
00511     * PPM/PPT marker segment parameters.
00512     \*****/
00513
00514     /* PPM marker segment parameters. */
00515
00516     typedef struct {
00517
00518         /* The index. */
00519         uint_fast8_t ind;
00520
00521         /* The length. */
00522         uint_fast16_t len;
00523
00524         /* The data. */
00525         jas_uchar *data;
00526
00527     } jpc_ppm_t;
00528

```



```

00529 /* PPT marker segment parameters. */
00530
00531 typedef struct {
00532
00533     /* The index. */
00534     uint_fast8_t ind;
00535
00536     /* The length. */
00537     uint_fast32_t len;
00538
00539     /* The data. */
00540     unsigned char *data;
00541 } jpc_ppt_t;
00542
00543 /*****\
00544 * COM marker segment parameters.
00545 \*****/
00546
00547 /*
00548  * Registration IDs.
00549  */
00550
00551 #define JPC_COM_BIN          0x00
00552 #define JPC_COM_LATIN       0x01
00553
00554 typedef struct {
00555
00556     /* The registration ID. */
00557     uint_fast16_t regid;
00558
00559     /* The length of the data in bytes. */
00560     uint_fast16_t len;
00561
00562     /* The data. */
00563     jas_uchar *data;
00564 } jpc_com_t;
00565
00566 /*****\
00567 * SOP marker segment parameters.
00568 \*****/
00569
00570 typedef struct {
00571
00572     /* The sequence number. */
00573     uint_fast16_t seqno;
00574 } jpc_sop_t;
00575
00576 /*****\
00577 * CRG marker segment parameters.
00578 \*****/
00579
00580 /* Per component information. */
00581
00582 typedef struct {
00583
00584     /* The horizontal offset. */
00585     uint_fast16_t hoff;
00586
00587     /* The vertical offset. */
00588     uint_fast16_t voff;
00589 } jpc_crgcomp_t;
00590
00591 typedef struct {
00592
00593     /* The number of components. */
00594     int numcomps;
00595
00596     /* Per component information. */
00597     jpc_crgcomp_t *comps;
00598 } jpc_crg_t;
00599
00600 /*****\
00601 * Marker segment parameters for unknown marker type.
00602 \*****/
00603
00604 typedef struct {

```

```

00610
00611     /* The data. */
00612     jas_uchar *data;
00613
00614     /* The length. */
00615     uint_fast16_t len;
00616
00617 } jpc_unk_t;
00618
00619 /*****\
00620 * Generic marker segment parameters.
00621 \*****/
00622
00623 typedef union {
00624     int soc;           /* unused */
00625     jpc_sot_t sot;
00626     int sod;           /* unused */
00627     int eoc;           /* unused */
00628     jpc_siz_t siz;
00629     jpc_cod_t cod;
00630     jpc_coc_t coc;
00631     jpc_rgn_t rgn;
00632     jpc_qcd_t qcd;
00633     jpc_qcc_t qcc;
00634     jpc_poc_t poc;
00635     /* jpc_plm_t plm; */
00636     /* jpc_plt_t plt; */
00637     jpc_ppm_t ppm;
00638     jpc_ppt_t ppt;
00639     jpc_sop_t sop;
00640     int eph;           /* unused */
00641     jpc_com_t com;
00642     jpc_crg_t crg;
00643     jpc_unk_t unk;
00644 } jpc_msparms_t;
00645
00646 /*****\
00647 * Marker segment.
00648 \*****/
00649
00650 /* Marker segment IDs. */
00651
00652 /* The smallest valid marker value. */
00653 #define JPC_MS_MIN      0xff00
00654
00655 /* The largest valid marker value. */
00656 #define JPC_MS_MAX      0xffff
00657
00658 /* The minimum marker value that cannot occur within packet data. */
00659 #define JPC_MS_INMIN    0xff80
00660 /* The maximum marker value that cannot occur within packet data. */
00661 #define JPC_MS_INMAX    0xffff
00662
00663 /* Delimiting marker segments. */
00664 #define JPC_MS_SOC      0xff4f /* Start of code stream (SOC). */
00665 #define JPC_MS_SOT      0xff90 /* Start of tile-part (SOT). */
00666 #define JPC_MS_SOD      0xff93 /* Start of data (SOD). */
00667 #define JPC_MS_EOC      0xffd9 /* End of code stream (EOC). */
00668
00669 /* Fixed information marker segments. */
00670 #define JPC_MS_SIZ      0xff51 /* Image and tile size (SIZ). */
00671
00672 /* Functional marker segments. */
00673 #define JPC_MS_COD      0xff52 /* Coding style default (COD). */
00674 #define JPC_MS_COC      0xff53 /* Coding style component (COC). */
00675 #define JPC_MS_RGN      0xff5e /* Region of interest (RGN). */
00676 #define JPC_MS_QCD      0xff5c /* Quantization default (QCD). */
00677 #define JPC_MS_QCC      0xff5d /* Quantization component (QCC). */
00678 #define JPC_MS_POC      0xff5f /* Progression order default (POC). */
00679
00680 /* Pointer marker segments. */
00681 #define JPC_MS_TLM      0xff55 /* Tile-part lengths, main header (TLM). */
00682 #define JPC_MS_PLM      0xff57 /* Packet length, main header (PLM). */
00683 #define JPC_MS_PLT      0xff58 /* Packet length, tile-part header (PLT). */
00684 #define JPC_MS_PPM      0xff60 /* Packed packet headers, main header (PPM). */
00685 #define JPC_MS_PPT      0xff61 /* Packet packet headers, tile-part header (PPT). */
00686
00687 /* In bit stream marker segments. */
00688 #define JPC_MS_SOP      0xff91 /* Start of packet (SOP). */
00689 #define JPC_MS_EPH      0xff92 /* End of packet header (EPH). */
00690

```

```

00691 /* Informational marker segments. */
00692 #define JPC_MS_CRG      0xff63 /* Component registration (CRG). */
00693 #define JPC_MS_COM      0xff64 /* Comment (COM). */
00694
00695 /* Forward declaration. */
00696 struct jpc_msops_s;
00697
00698 /* Generic marker segment class. */
00699
00700 typedef struct {
00701
00702     /* The type of marker segment. */
00703     uint_fast16_t id;
00704
00705     /* The length of the marker segment. */
00706     uint_fast16_t len;
00707
00708     /* The starting offset within the stream. */
00709     uint_fast32_t off;
00710
00711     /* The parameters of the marker segment. */
00712     jpc_msparms_t parms;
00713
00714     /* The marker segment operations. */
00715     const struct jpc_msops_s *ops;
00716 } jpc_ms_t;
00717
00718 /* Marker segment operations (which depend on the marker segment type). */
00719
00720 typedef struct jpc_msops_s {
00721
00722     /* Destroy the marker segment parameters. */
00723     void (*destroyparms)(jpc_ms_t *ms);
00724
00725     /* Get the marker segment parameters from a stream. */
00726     int (*getparms)(jpc_ms_t *ms, jpc_cstate_t *cstate, jas_stream_t *in);
00727
00728     /* Put the marker segment parameters to a stream. */
00729     int (*putparms)(jpc_ms_t *ms, jpc_cstate_t *cstate, jas_stream_t *out);
00730
00731     /* Dump the marker segment parameters (for debugging). */
00732     int (*dumpparms)(jpc_ms_t *ms);
00733 } jpc_msops_t;
00734
00735
00736
00737 /*****
00738  * Macros/Functions.
00739  *****/
00740
00741 /* Create a code-stream state object. */
00742 jpc_cstate_t *jpc_cstate_create(void);
00743
00744 /* Destroy a code-stream state object. */
00745 void jpc_cstate_destroy(jpc_cstate_t *cstate);
00746
00747 /* Create a marker segment. */
00748 jpc_ms_t *jpc_ms_create(int type);
00749
00750 /* Destroy a marker segment. */
00751 void jpc_ms_destroy(jpc_ms_t *ms);
00752
00753 /* Does a marker segment have parameters? */
00754 JAS_ATTRIBUTE_CONST
00755 static inline bool JPC_MS_HASPARMS(unsigned x)
00756 {
00757     return !(x == JPC_MS_SOC || x == JPC_MS_SOD || x == JPC_MS_EOC ||
00758             x == JPC_MS_EPH || (x >= 0xff30 && x <= 0xff3f));
00759 }
00760
00761 /* Get the marker segment type. */
00762 JAS_ATTRIBUTE_PURE
00763 static inline unsigned jpc_ms_gettype(const jpc_ms_t *ms)
00764 {
00765     return ms->id;
00766 }
00767
00768 /* Read a marker segment from a stream. */
00769 jpc_ms_t *jpc_getms(jas_stream_t *in, jpc_cstate_t *cstate);
00770
00771 /* Write a marker segment to a stream. */

```

```

00772 int jpc_putms(jas_stream_t *out, jpc_cstate_t *cstate, jpc_ms_t *ms);
00773
00774 /* Copy code stream data from one stream to another. */
00775 int jpc_getdata(jas_stream_t *in, jas_stream_t *out, long n);
00776
00777 /* Copy code stream data from one stream to another. */
00778 int jpc_putdata(jas_stream_t *out, jas_stream_t *in, long n);
00779
00780 /* Dump a marker segment (for debugging). */
00781 void jpc_ms_dump(jpc_ms_t *ms);
00782
00783 /* Read a 8-bit unsigned integer from a stream. */
00784 int jpc_getuint8(jas_stream_t *in, uint_fast8_t *val);
00785
00786 /* Read a 16-bit unsigned integer from a stream. */
00787 int jpc_getuint16(jas_stream_t *in, uint_fast16_t *val);
00788
00789 /* Read a 32-bit unsigned integer from a stream. */
00790 int jpc_getuint32(jas_stream_t *in, uint_fast32_t *val);
00791
00792 /* Write a 8-bit unsigned integer to a stream. */
00793 int jpc_putuint8(jas_stream_t *out, uint_fast8_t val);
00794
00795 /* Write a 16-bit unsigned integer to a stream. */
00796 int jpc_putuint16(jas_stream_t *out, uint_fast16_t val);
00797
00798 /* Write a 32-bit unsigned integer to a stream. */
00799 int jpc_putuint32(jas_stream_t *out, uint_fast32_t val);
00800
00801 #endif

```

16.50 jpc_dec.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION

```

```

00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00064  /*
00065  * JPEG-2000 Decoder
00066  *
00067  * $Id$
00068  */
00069
00070  #ifndef JPC_DEC_H
00071  #define JPC_DEC_H
00072
00073  /*****\
00074  * Includes.
00075  \*****/
00076
00077  #include "jasper/jas_stream.h"
00078  #include "jasper/jas_image.h"
00079
00080  #include "jpc_cod.h"
00081  #include "jpc_tsfb.h"
00082  #include "jpc_tagtree.h"
00083  #include "jpc_cs.h"
00084  #include "jpc_tlcod.h"
00085  #include "jpc_t2cod.h"
00086
00087  /*****\
00088  * Below are some ugly warts necessary to support packed packet headers.
00089  \*****/
00090
00091  /* PPM/PPT marker segment table entry. */
00092
00093  typedef struct {
00094
00095      /* The index for this entry. */
00096      uint_fast16_t ind;
00097
00098      /* The data length. */
00099      uint_fast32_t len;
00100
00101      /* The data. */
00102      jas_uchar *data;
00103  } jpc_ppxstabent_t;
00104
00105  /* PPM/PPT marker segment table. */
00106
00107  typedef struct {
00108
00109      /* The number of entries. */
00110      unsigned numents;
00111
00112      /* The maximum number of entries (i.e., the allocated size of the array
00113      below). */
00114      unsigned maxents;
00115
00116      /* The table entries. */
00117      jpc_ppxstabent_t **ents;
00118  } jpc_ppxstab_t;
00119
00120  /* Stream list class. */
00121
00122  typedef struct {

```

```

00125
00126     /* The number of streams in this list. */
00127     unsigned numstreams;
00128
00129     /* The maximum number of streams that can be accomodated without
00130        growing the streams array. */
00131     unsigned maxstreams;
00132
00133     /* The streams. */
00134     jas_stream_t **streams;
00135
00136 } jpc_streamlist_t;
00137
00138 /*****
00139  * Coding parameters class.
00140  \*****/
00141
00142 /* Per-component coding parameters. */
00143
00144 typedef struct {
00145
00146     /* How were various coding parameters set? */
00147     unsigned flags;
00148
00149     /* Per-component coding style parameters (e.g., explicit precinct sizes) */
00150     uint_fast8_t csty;
00151
00152     /* The number of resolution levels. */
00153     uint_fast8_t numrlvls;
00154
00155     /* The code block width exponent. */
00156     uint_fast8_t cblkwidthexpn;
00157
00158     /* The code block height exponent. */
00159     uint_fast8_t cblkheightexpn;
00160
00161     /* The QMFB ID. */
00162     uint_fast8_t qmfbid;
00163
00164     /* The quantization style. */
00165     uint_fast8_t qsty;
00166
00167     /* The number of quantizer step sizes. */
00168     uint_fast16_t numstepsizes;
00169
00170     /* The step sizes. */
00171     uint_fast16_t stepsizes[3 * JPC_MAXRLVLS + 1];
00172
00173     /* The number of guard bits. */
00174     uint_fast8_t numguardbits;
00175
00176     /* The ROI shift value. */
00177     uint_fast8_t roishift;
00178
00179     /* The code block parameters. */
00180     uint_fast8_t cblkctx;
00181
00182     /* The precinct width exponents. */
00183     uint_fast8_t prcwidthexpns[JPC_MAXRLVLS];
00184
00185     /* The precinct height exponents. */
00186     uint_fast8_t prcheightexpns[JPC_MAXRLVLS];
00187
00188 } jpc_dec_ccp_t;
00189
00190 /* Coding parameters. */
00191
00192 typedef struct {
00193
00194     /* How were these coding parameters set? */
00195     unsigned flags;
00196
00197     /* Progression change list. */
00198     jpc_pchglst_t *pchglst;
00199
00200     /* Progression order. */
00201     uint_fast8_t prgord;
00202
00203     /* The number of layers. */
00204     uint_fast16_t numlyrs;
00205

```

```

00206      /* The MCT ID. */
00207      uint_fast8_t mctid;
00208
00209      /* The coding style parameters (e.g., SOP, EPH). */
00210      uint_fast8_t csty;
00211
00212      /* The number of components. */
00213      unsigned numcomps;
00214
00215      /* The per-component coding parameters. */
00216      jpc_dec_ccp_t *ccps;
00217
00218 } jpc_dec_cp_t;
00219
00220 /*****
00221  * Decoder class.
00222  *****/
00223
00224 /* Decoder per-segment state information. */
00225
00226 typedef struct jpc_dec_seg_s {
00227
00228     /* The next segment in the list. */
00229     struct jpc_dec_seg_s *next;
00230
00231     /* The previous segment in the list. */
00232     struct jpc_dec_seg_s *prev;
00233
00234     /* The starting pass number for this segment. */
00235     unsigned passno;
00236
00237     /* The number of passes in this segment. */
00238     unsigned numpasses;
00239
00240     /* The maximum number of passes in this segment. */
00241     unsigned maxpasses;
00242
00243     /* The type of data in this segment (i.e., MQ or raw). */
00244     enum jpc_segtype type;
00245
00246     /* A stream containing the data for this segment. */
00247     jas_stream_t *stream;
00248
00249     /* The number of bytes destined for this segment from the packet
00250        currently being decoded. */
00251     unsigned cnt;
00252
00253     /* A flag indicating if this segment has been terminated. */
00254     int complete;
00255
00256     /* The layer number to which this segment belongs. */
00257     /* If the segment spans multiple layers, then the largest layer number
00258        spanned by the segment is used. */
00259     unsigned lyrno;
00260
00261 } jpc_dec_seg_t;
00262
00263 /* Decoder segment list. */
00264
00265 typedef struct {
00266
00267     /* The first entry in the list. */
00268     jpc_dec_seg_t *head;
00269
00270     /* The last entry in the list. */
00271     jpc_dec_seg_t *tail;
00272
00273 } jpc_dec_seglist_t;
00274
00275 /* Decoder per-code-block state information. */
00276
00277 typedef struct {
00278
00279     /* The number of passes. */
00280     unsigned numpasses;
00281
00282     /* A list of segments that still need to be decoded. */
00283     jpc_dec_seglist_t segs;
00284
00285     /* The first incomplete/partial segment. */
00286     jpc_dec_seg_t *curseg;

```

```

00287
00288     /* The number of leading insignificant bit planes for this code block. */
00289     unsigned numimsbs;
00290
00291     /* The number of bits used to encode pass data lengths. */
00292     unsigned numlenbits;
00293
00294     /* The first pass number containing data for this code block. */
00295     unsigned firstpassno;
00296
00297     /* The sample data associated with this code block. */
00298     jas_matrix_t *data;
00299 } jpc_dec_cblk_t;
00300
00301 /* Decoder per-code-block-group state information. */
00302
00303 typedef struct {
00304     /* The x-coordinate of the top-left corner of the precinct. */
00305     uint_fast32_t xstart;
00306
00307     /* The y-coordinate of the top-left corner of the precinct. */
00308     uint_fast32_t ystart;
00309
00310     /* The x-coordinate of the bottom-right corner of the precinct
00311        (plus one). */
00312     uint_fast32_t xend;
00313
00314     /* The y-coordinate of the bottom-right corner of the precinct
00315        (plus one). */
00316     uint_fast32_t yend;
00317
00318     /* The number of code blocks spanning this precinct in the horizontal
00319        direction. */
00320     unsigned numhcbcls;
00321
00322     /* The number of code blocks spanning this precinct in the vertical
00323        direction. */
00324     unsigned numvcblks;
00325
00326     /* The total number of code blocks in this precinct. */
00327     unsigned numcblks;
00328
00329     /* The per code block information. */
00330     jpc_dec_cblk_t *cblks;
00331
00332     /* The inclusion tag tree. */
00333     jpc_tagtree_t *incltagtree;
00334
00335     /* The insignificant MSBs tag tree. */
00336     jpc_tagtree_t *numimsbstagtree;
00337 } jpc_dec_prc_t;
00338
00339 /* Decoder per-band state information. */
00340
00341 typedef struct {
00342     /* The per-code-block-group state information. */
00343     jpc_dec_prc_t *prcs;
00344
00345     /* The sample data associated with this band. */
00346     jas_matrix_t *data;
00347
00348     /* The orientation of this band (i.e., LL, LH, HL, or HH). */
00349     enum jpc_tsfb_orient orient;
00350
00351     /* The encoded quantizer step size. */
00352     unsigned stepsize;
00353
00354     /* The absolute quantizer step size. */
00355     jpc_fix_t absstepsize;
00356
00357     /* The number of bit planes for this band. */
00358     unsigned numbps;
00359
00360     /* The analysis gain associated with this band. */
00361     int analgain;
00362
00363     /* The ROI shift value for this band. */

```



```

00368         int roishift;
00369
00370     } jpc_dec_band_t;
00371
00372     /* Decoder per-resolution-level state information. */
00373
00374     typedef struct {
00375
00376         /* The number of bands associated with this resolution level. */
00377         unsigned numbands;
00378
00379         /* The per-band information. */
00380         jpc_dec_band_t *bands;
00381
00382         /* The x-coordinate of the top-left corner of the tile-component
00383            at this resolution. */
00384         uint_fast32_t xstart;
00385
00386         /* The y-coordinate of the top-left corner of the tile-component
00387            at this resolution. */
00388         uint_fast32_t ystart;
00389
00390         /* The x-coordinate of the bottom-right corner of the tile-component
00391            at this resolution (plus one). */
00392         uint_fast32_t xend;
00393
00394         /* The y-coordinate of the bottom-right corner of the tile-component
00395            at this resolution (plus one). */
00396         uint_fast32_t yend;
00397
00398         /* The exponent value for the nominal precinct width measured
00399            relative to the associated LL band. */
00400         unsigned prcwidthexpn;
00401
00402         /* The exponent value for the nominal precinct height measured
00403            relative to the associated LL band. */
00404         unsigned prcheightexpn;
00405
00406         /* The number of precincts in the horizontal direction. */
00407         unsigned numhprcs;
00408
00409         /* The number of precincts in the vertical direction. */
00410         unsigned numvprcs;
00411
00412         /* The total number of precincts. */
00413         unsigned numprcs;
00414
00415         /* The exponent value for the nominal code block group width.
00416            This quantity is associated with the next lower resolution level
00417            (assuming that there is one). */
00418         unsigned cbgwidthexpn;
00419
00420         /* The exponent value for the nominal code block group height
00421            This quantity is associated with the next lower resolution level
00422            (assuming that there is one). */
00423         unsigned cbgheightexpn;
00424
00425         /* The exponent value for the code block width. */
00426         uint_fast16_t cblkwidthexpn;
00427
00428         /* The exponent value for the code block height. */
00429         uint_fast16_t cblkheightexpn;
00430
00431     } jpc_dec_rlvl_t;
00432
00433     /* Decoder per-tile-component state information. */
00434
00435     typedef struct {
00436
00437         /* The x-coordinate of the top-left corner of the tile-component
00438            in the coordinate system of the tile-component. */
00439         uint_fast32_t xstart;
00440
00441         /* The y-coordinate of the top-left corner of the tile-component
00442            in the coordinate system of the tile-component. */
00443         uint_fast32_t ystart;
00444
00445         /* The x-coordinate of the bottom-right corner of the tile-component
00446            in the coordinate system of the tile-component (plus one). */
00447         uint_fast32_t xend;
00448

```

```

00449      /* The y-coordinate of the bottom-right corner of the tile-component
00450      in the coordinate system of the tile-component (plus one). */
00451      uint_fast32_t yend;
00452
00453      /* The component data for the current tile. */
00454      jas_matrix_t *data;
00455
00456      /* The number of resolution levels. */
00457      unsigned numrlvls;
00458
00459      /* The per resolution level information. */
00460      jpc_dec_rlvl_t *rlvls;
00461
00462      /* The TSFB. */
00463      jpc_tsfb_t *tsfb;
00464
00465  } jpc_dec_tcomp_t;
00466
00467  /*
00468   * Tile states.
00469   */
00470
00471  #define JPC_TILE_INIT    0
00472  #define JPC_TILE_ACTIVE  1
00473  #define JPC_TILE_ACTIVELAST  2
00474  #define JPC_TILE_DONE    3
00475
00476  /* Decoder per-tile state information. */
00477
00478  typedef struct {
00479
00480      /* The processing state for this tile. */
00481      int state;
00482
00483      /* The x-coordinate of the top-left corner of the tile on the reference
00484      grid. */
00485      uint_fast32_t xstart;
00486
00487      /* The y-coordinate of the top-left corner of the tile on the reference
00488      grid. */
00489      uint_fast32_t ystart;
00490
00491      /* The x-coordinate of the bottom-right corner of the tile on the
00492      reference grid (plus one). */
00493      uint_fast32_t xend;
00494
00495      /* The y-coordinate of the bottom-right corner of the tile on the
00496      reference grid (plus one). */
00497      uint_fast32_t yend;
00498
00499      /* The packed packet header data for this tile. */
00500      jpc_ppxstab_t *pptstab;
00501
00502      /* A stream containing the packed packet header data for this tile. */
00503      jas_stream_t *pkthdrstream;
00504
00505      /* The coding parameters for this tile. */
00506      jpc_dec_cp_t *cp;
00507
00508      /* The per tile-component information. */
00509      jpc_dec_tcomp_t *tcomps;
00510
00511      /* The next expected tile-part number. */
00512      unsigned partno;
00513
00514      /* The number of tile-parts. */
00515      unsigned numparts;
00516
00517      /* The coding mode. */
00518      int realmode;
00519
00520      /* The packet iterator for this tile. */
00521      jpc_pi_t *pi;
00522  } jpc_dec_tile_t;
00523
00524
00525  /* Decoder per-component state information. */
00526
00527  typedef struct {
00528
00529      /* The horizontal sampling period. */

```

```

00530     uint_fast32_t hstep;
00531
00532     /* The vertical sampling period. */
00533     uint_fast32_t vstep;
00534
00535     /* The number of samples in the horizontal direction. */
00536     uint_fast32_t width;
00537
00538     /* The number of samples in the vertical direction. */
00539     uint_fast32_t height;
00540
00541     /* The precision of the sample data. */
00542     uint_fast16_t prec;
00543
00544     /* The signedness of the sample data. */
00545     bool sgnd;
00546
00547     /* The sample alignment horizontal offset. */
00548     uint_fast32_t hsubstep;
00549
00550     /* The sample alignment vertical offset. */
00551     uint_fast32_t vsubstep;
00552
00553 } jpc_dec_cmpt_t;
00554
00555 /* Decoder state information. */
00556
00557 typedef struct {
00558
00559     /* The decoded image. */
00560     jas_image_t *image;
00561
00562     /* The x-coordinate of the top-left corner of the image area on
00563        the reference grid. */
00564     uint_fast32_t xstart;
00565
00566     /* The y-coordinate of the top-left corner of the image area on
00567        the reference grid. */
00568     uint_fast32_t ystart;
00569
00570     /* The x-coordinate of the bottom-right corner of the image area on
00571        the reference grid (plus one). */
00572     uint_fast32_t xend;
00573
00574     /* The y-coordinate of the bottom-right corner of the image area on
00575        the reference grid (plus one). */
00576     uint_fast32_t yend;
00577
00578     /* The nominal tile width in units of the image reference grid. */
00579     uint_fast32_t tilewidth;
00580
00581     /* The nominal tile height in units of the image reference grid. */
00582     uint_fast32_t tileheight;
00583
00584     /* The horizontal offset from the origin of the reference grid to the
00585        left side of the first tile. */
00586     uint_fast32_t tilexoff;
00587
00588     /* The vertical offset from the origin of the reference grid to the
00589        top side of the first tile. */
00590     uint_fast32_t tileyoff;
00591
00592     /* The number of tiles spanning the image area in the vertical
00593        direction. */
00594     unsigned numhtiles;
00595
00596     /* The number of tiles spanning the image area in the horizontal
00597        direction. */
00598     unsigned numvtiles;
00599
00600     /* The total number of tiles. */
00601     unsigned numtiles;
00602
00603     /* The per-tile information. */
00604     jpc_dec_tile_t *tiles;
00605
00606     /* The tile currently being processed. */
00607     jpc_dec_tile_t *curtile;
00608
00609     /* The number of components. */
00610     unsigned numcomps;

```

```

00611
00612     /* The stream containing the input JPEG-2000 code stream data. */
00613     jas_stream_t *in;
00614
00615     /* The default coding parameters for all tiles. */
00616     jpc_dec_cp_t *cp;
00617
00618     /* The maximum number of layers that may be decoded. */
00619     unsigned maxlyrs;
00620
00621     /* The maximum number of packets that may be decoded. */
00622     int maxpkts;
00623
00624     /* The number of packets decoded so far in the processing of the entire
00625        code stream. */
00626     unsigned numpkts;
00627
00628     /* The next expected PPM marker segment sequence number. */
00629     unsigned ppmseqno;
00630
00631     /* The current state for code stream processing. */
00632     int state;
00633
00634     /* The per-component information. */
00635     jpc_dec_cmpt_t *cmpts;
00636
00637     /* The information from PPM marker segments. */
00638     jpc_ppxstab_t *ppmstab;
00639
00640     /* A list of streams containing packet header data from PPM marker
00641        segments. */
00642     jpc_streamlist_t *pkthdrstreams;
00643
00644     /* The expected ending offset for a tile-part. */
00645     long curtileendoff;
00646
00647     /* This is required by the tier-2 decoder. */
00648     jpc_cstate_t *cstate;
00649
00650     size_t max_samples;
00651
00652 } jpc_dec_t;
00653
00654 /* Decoder options. */
00655
00656 typedef struct {
00657
00658     /* The debug level for the decoder. */
00659     int debug;
00660
00661     /* The maximum number of layers to decode. */
00662     unsigned maxlyrs;
00663
00664     /* The maximum number of packets to decode. */
00665     int maxpkts;
00666
00667     size_t max_samples;
00668
00669 } jpc_dec_importopts_t;
00670
00671 /*****
00672  * Functions.
00673  *****/
00674
00675 /* Create a decoder segment object. */
00676 jpc_dec_seg_t *jpc_seg_alloc(void);
00677
00678 /* Destroy a decoder segment object. */
00679 void jpc_seg_destroy(jpc_dec_seg_t *seg);
00680
00681 /* Remove a segment from a segment list. */
00682 void jpc_seglist_remove(jpc_dec_seglist_t *list, jpc_dec_seg_t *node);
00683
00684 /* Insert a segment into a segment list. */
00685 void jpc_seglist_insert(jpc_dec_seglist_t *list, jpc_dec_seg_t *ins,
00686     jpc_dec_seg_t *node);
00687
00688 #endif

```

16.51 jpc_enc.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00064 /*
00065  * $Id$
00066  */
00067
00068 #ifndef JPC_ENC_H
00069 #define JPC_ENC_H
00070
00071 /*****\
00072  * Includes.
00073  \*****/
00074
00075 #include "jasper/jas_image.h"
00076 #include "jasper/jas_seq.h"
00077 #include "jasper/jas_stream.h"
00078

```

```

00079 #include "jpc_cod.h"
00080 #include "jpc_t1cod.h"
00081 #include "jpc_t2cod.h"
00082 #include "jpc_mqenc.h"
00083 #include "jpc_tagtree.h"
00084 #include "jpc_cs.h"
00085 #include "jpc_fix.h"
00086 #include "jpcflt.h"
00087 #include "jpc_tsfb.h"
00088
00089 /*****
00090  * Constants.
00091  *****/
00092
00093 /* The number of bits used in various lookup tables. */
00094 #define JPC_NUMEXTRABITS      JPC_NMSEDEC_FRACBITS
00095
00096 /* An invalid R-D slope value. */
00097 #define JPC_BADRDSLOPE      (-1)
00098
00099 /*****
00100  * Coding parameters types.
00101  *****/
00102
00103 /* Per-component coding paramters. */
00104
00105 typedef struct {
00106
00107     /* The horizontal sampling period. */
00108     uint_fast8_t sampgrdstepx;
00109
00110     /* The vertical sampling period. */
00111     uint_fast8_t sampgrdstepy;
00112
00113     /* The sample alignment horizontal offset. */
00114     uint_fast8_t sampgrdsstepx;
00115
00116     /* The sample alignment vertical offset. */
00117     uint_fast8_t sampgrdsstepy;
00118
00119     /* The precision of the samples. */
00120     uint_fast8_t prec;
00121
00122     /* The signedness of the samples. */
00123     bool sgnd;
00124
00125     /* The number of step sizes. */
00126     uint_fast16_t numstepsizes;
00127
00128     /* The quantizer step sizes. */
00129     uint_fast16_t stepsizes[JPC_MAXBANDS];
00130
00131 } jpc_enc_ccp_t;
00132
00133 /* Per-tile coding parameters. */
00134
00135 typedef struct {
00136
00137     /* The coding mode. */
00138     bool intmode;
00139
00140     /* The coding style (i.e., SOP, EPH). */
00141     uint_fast8_t csty;
00142
00143     /* The progression order. */
00144     uint_fast8_t prg;
00145
00146     /* The multicomponent transform. */
00147     uint_fast8_t mctid;
00148
00149     /* The number of layers. */
00150     uint_fast16_t numlyrs;
00151
00152     /* The normalized bit rates associated with the various
00153        intermediate layers. */
00154     jpc_fix_t *ilyrrates;
00155
00156 } jpc_enc_tcp_t;
00157
00158 /* Per tile-component coding parameters. */
00159

```

```

00160 typedef struct {
00161
00162     /* The coding style (i.e., explicit precinct sizes). */
00163     uint_fast8_t csty;
00164
00165     /* The maximum number of resolution levels allowed. */
00166     uint_fast8_t maxrlvls;
00167
00168     /* The exponent for the nominal code block width. */
00169     uint_fast16_t cblkwidthexpn;
00170
00171     /* The exponent for the nominal code block height. */
00172     uint_fast16_t cblkheightexpn;
00173
00174     /* The code block style parameters (e.g., lazy, terminate all,
00175      segmentation symbols, causal, reset probability models). */
00176     uint_fast8_t cblksty;
00177
00178     /* The QMFB. */
00179     uint_fast8_t qmfbid;
00180
00181     /* The precinct width values. */
00182     uint_fast16_t prcwidthexpns[JPC_MAXRLVLS];
00183
00184     /* The precinct height values. */
00185     uint_fast16_t prcheightexpns[JPC_MAXRLVLS];
00186
00187     /* The number of guard bits. */
00188     uint_fast8_t numgbits;
00189
00190 } jpc_enc_tccp_t;
00191
00192 /* Coding parameters. */
00193 typedef struct {
00194
00195     /* The debug level. */
00196     int debug;
00197
00198     /* The horizontal offset from the origin of the reference grid to the
00199      left edge of the image area. */
00200     uint_fast32_t imgareatl;
00201
00202     /* The vertical offset from the origin of the reference grid to the
00203      top edge of the image area. */
00204     uint_fast32_t imgareat;
00205
00206     /* The horizontal offset from the origin of the reference grid to the
00207      right edge of the image area (plus one). */
00208     uint_fast32_t refgrdwidth;
00209
00210     /* The vertical offset from the origin of the reference grid to the
00211      bottom edge of the image area (plus one). */
00212     uint_fast32_t refgrdheight;
00213
00214     /* The horizontal offset from the origin of the tile grid to the
00215      origin of the reference grid. */
00216     uint_fast32_t tilegrdoffx;
00217
00218     /* The vertical offset from the origin of the tile grid to the
00219      origin of the reference grid. */
00220     uint_fast32_t tilegrdoffy;
00221
00222     /* The nominal tile width in units of the image reference grid. */
00223     uint_fast32_t tilewidth;
00224
00225     /* The nominal tile height in units of the image reference grid. */
00226     uint_fast32_t tileheight;
00227
00228     /* The number of tiles spanning the image area in the horizontal
00229      direction. */
00230     uint_fast32_t numhtiles;
00231
00232     /* The number of tiles spanning the image area in the vertical
00233      direction. */
00234     uint_fast32_t numvtiles;
00235
00236     /* The number of tiles. */
00237     uint_fast32_t numtiles;
00238
00239     /* The number of components. */
00240

```

```

00241     uint_fast16_t numcmpts;
00242
00243     /* The per-component coding parameters. */
00244     jpc_enc_ccp_t *ccps;
00245
00246     /* The per-tile coding parameters. */
00247     jpc_enc_tcp_t tcp;
00248
00249     /* The per-tile-component coding parameters. */
00250     jpc_enc_tccp_t tccp;
00251
00252     /* The target code stream length in bytes. */
00253     uint_fast32_t totalsize;
00254
00255     /* The raw (i.e., uncompressed) size of the image in bytes. */
00256     uint_fast32_t rawsize;
00257
00258 } jpc_enc_cp_t;
00259
00260 /*****
00261  * Encoder class.
00262  *****/
00263
00264 /* Encoder per-coding-pass state information. */
00265
00266 typedef struct {
00267
00268     /* The starting offset for this pass. */
00269     int start;
00270
00271     /* The ending offset for this pass. */
00272     int end;
00273
00274     /* The type of data in this pass (i.e., MQ or raw). */
00275     enum jpc_segtype type;
00276
00277     /* Flag indicating that this pass is terminated. */
00278     int term;
00279
00280     /* The entropy coder state after coding this pass. */
00281     jpc_mqencstate_t mqencstate;
00282
00283     /* The layer to which this pass has been assigned. */
00284     unsigned lyrno;
00285
00286     /* The R-D slope for this pass. */
00287     jpc_flt_t rdslope;
00288
00289     /* The weighted MSE reduction associated with this pass. */
00290     jpc_flt_t wmsedec;
00291
00292     /* The cumulative weighted MSE reduction. */
00293     jpc_flt_t cumwmsedec;
00294
00295     /* The normalized MSE reduction. */
00296     long nmsedec;
00297
00298 } jpc_enc_pass_t;
00299
00300 /* Encoder per-code-block state information. */
00301
00302 typedef struct {
00303
00304     /* The number of passes. */
00305     unsigned numpasses;
00306
00307     /* The per-pass information. */
00308     jpc_enc_pass_t *passes;
00309
00310     /* The number of passes encoded so far. */
00311     int numencpasses;
00312
00313     /* The number of insignificant MSBs. */
00314     int numimsbs;
00315
00316     /* The number of bits used to encode pass data lengths. */
00317     int numlenbits;
00318
00319     /* The byte stream for this code block. */
00320     jas_stream_t *stream;
00321

```



```

00322     /* The entropy encoder. */
00323     jpc_mqenc_t *mqenc;
00324
00325     /* The data for this code block. */
00326     jas_matrix_t *data;
00327
00328     /* The state for this code block. */
00329     jas_matrix_t *flags;
00330
00331     /* The number of bit planes required for this code block. */
00332     int numbps;
00333
00334     /* The next pass to be encoded. */
00335     jpc_enc_pass_t *curpass;
00336
00337     /* The per-code-block-group state information. */
00338     struct jpc_enc_prc_s *prc;
00339
00340     /* The saved current pass. */
00341     /* This is used by the rate control code. */
00342     jpc_enc_pass_t *savedcurpass;
00343
00344     /* The saved length indicator size. */
00345     /* This is used by the rate control code. */
00346     int savednumlenbits;
00347
00348     /* The saved number of encoded passes. */
00349     /* This is used by the rate control code. */
00350     int savednumencpasses;
00351
00352 } jpc_enc_cblk_t;
00353
00354 /* Encoder per-code-block-group state information. */
00355
00356 typedef struct jpc_enc_prc_s {
00357
00358     /* The x-coordinate of the top-left corner of the precinct. */
00359     uint_fast32_t tlx;
00360
00361     /* The y-coordinate of the top-left corner of the precinct. */
00362     uint_fast32_t tly;
00363
00364     /* The x-coordinate of the bottom-right corner of the precinct
00365        (plus one). */
00366     uint_fast32_t brx;
00367
00368     /* The y-coordinate of the bottom-right corner of the precinct
00369        (plus one). */
00370     uint_fast32_t bry;
00371
00372     /* The number of code blocks spanning the precinct in the horizontal
00373        direction. */
00374     int numhcbks;
00375
00376     /* The number of code blocks spanning the precinct in the vertical
00377        direction. */
00378     int numvcblks;
00379
00380     /* The total number of code blocks. */
00381     unsigned numcblks;
00382
00383     /* The per-code-block information. */
00384     jpc_enc_cblk_t *cblks;
00385
00386     /* The inclusion tag tree. */
00387     jpc_tagtree_t *incltree;
00388
00389     /* The insignificant MSBs tag tree. */
00390     jpc_tagtree_t *nlibtree;
00391
00392     /* The per-band information. */
00393     struct jpc_enc_band_s *band;
00394
00395     /* The saved inclusion tag tree. */
00396     /* This is used by rate control. */
00397     jpc_tagtree_t *savincltree;
00398
00399     /* The saved leading-insignificant-bit-planes tag tree. */
00400     /* This is used by rate control. */
00401     jpc_tagtree_t *savnlibtree;
00402

```

```

00403 } jpc_enc_prc_t;
00404
00405 /* Encoder per-band state information. */
00406
00407 typedef struct jpc_enc_band_s {
00408
00409     /* The per precinct information. */
00410     jpc_enc_prc_t *prcs;
00411
00412     /* The coefficient data for this band. */
00413     jas_matrix_t *data;
00414
00415     /* The orientation of this band (i.e., LL, LH, HL, or HH). */
00416     enum jpc_tsfb_orient orient;
00417
00418     /* The number of bit planes associated with this band. */
00419     int numbps;
00420
00421     /* The quantizer step size. */
00422     jpc_fix_t absstepsize;
00423
00424     /* The encoded quantizer step size. */
00425     int stepsize;
00426
00427     /* The L2 norm of the synthesis basis functions associated with
00428        this band. (The MCT is not considered in this value.) */
00429     jpc_fix_t synweight;
00430
00431     /* The analysis gain for this band. */
00432     int analgain;
00433
00434     /* The per-resolution-level information. */
00435     struct jpc_enc_rlvl_s *rlvl;
00436 } jpc_enc_band_t;
00437
00438 /* Encoder per-resolution-level state information. */
00439
00440 typedef struct jpc_enc_rlvl_s {
00441
00442     /* The x-coordinate of the top-left corner of the tile-component
00443        at this resolution. */
00444     uint_fast32_t tlx;
00445
00446     /* The y-coordinate of the top-left corner of the tile-component
00447        at this resolution. */
00448     uint_fast32_t tly;
00449
00450     /* The x-coordinate of the bottom-right corner of the tile-component
00451        at this resolution (plus one). */
00452     uint_fast32_t brx;
00453
00454     /* The y-coordinate of the bottom-right corner of the tile-component
00455        at this resolution (plus one). */
00456     uint_fast32_t bry;
00457
00458     /* The exponent value for the nominal precinct width measured
00459        relative to the associated LL band. */
00460     int prcwidthexpn;
00461
00462     /* The exponent value for the nominal precinct height measured
00463        relative to the associated LL band. */
00464     int prcheightexpn;
00465
00466     /* The number of precincts spanning the resolution level in the
00467        horizontal direction. */
00468     int numhprcs;
00469
00470     /* The number of precincts spanning the resolution level in the
00471        vertical direction. */
00472     int numvprcs;
00473
00474     /* The total number of precincts. */
00475     unsigned numprcs;
00476
00477     /* The exponent value for the nominal code block group width.
00478        This quantity is associated with the next lower resolution level
00479        (assuming that there is one). */
00480     unsigned cbgwidthexpn;
00481
00482     /* The exponent value for the nominal code block group height.
00483

```

```

00484         This quantity is associated with the next lower resolution level
00485         (assuming that there is one). */
00486         unsigned cbgheightexpn;
00487
00488         /* The exponent value for the code block width. */
00489         uint_fast16_t cblkwidthexpn;
00490
00491         /* The exponent value for the code block height. */
00492         uint_fast16_t cblkheightexpn;
00493
00494         /* The number of bands associated with this resolution level. */
00495         unsigned numbands;
00496
00497         /* The per-band information. */
00498         jpc_enc_band_t *bands;
00499
00500         /* The parent tile-component. */
00501         struct jpc_enc_tcmpt_s *tcmpt;
00502
00503     } jpc_enc_rlvl_t;
00504
00505     /* Encoder per-tile-component state information. */
00506
00507     typedef struct jpc_enc_tcmpt_s {
00508
00509         /* The number of resolution levels. */
00510         unsigned numrlvls;
00511
00512         /* The per-resolution-level information. */
00513         jpc_enc_rlvl_t *rlvls;
00514
00515         /* The tile-component data. */
00516         jas_matrix_t *data;
00517
00518         /* The QMFB. */
00519         int qmfbid;
00520
00521         /* The number of bands. */
00522         int numbands;
00523
00524         /* The TSFB. */
00525         jpc_tsfb_t *tsfb;
00526
00527         /* The synthesis energy weight (for the MCT). */
00528         jpc_fix_t synweight;
00529
00530         /* The precinct width exponents. */
00531         int prcwidthexpns[JPC_MAXRLVLS];
00532
00533         /* The precinct height exponents. */
00534         int prcheightexpns[JPC_MAXRLVLS];
00535
00536         /* The code block width exponent. */
00537         int cblkwidthexpn;
00538
00539         /* The code block height exponent. */
00540         int cblkheightexpn;
00541
00542         /* Coding style (i.e., explicit precinct sizes). */
00543         int csty;
00544
00545         /* Code block style. */
00546         int cblksty;
00547
00548         /* The number of quantizer step sizes. */
00549         int numstepsizes;
00550
00551         /* The encoded quantizer step sizes. */
00552         uint_fast16_t stepsizes[JPC_MAXBANDS];
00553
00554         /* The parent tile. */
00555         struct jpc_enc_tile_s *tile;
00556
00557     } jpc_enc_tcmpt_t;
00558
00559     /* Encoder per-tile state information. */
00560
00561     typedef struct jpc_enc_tile_s {
00562
00563         /* The tile number. */
00564         uint_fast32_t tileno;

```

```

00565
00566     /* The x-coordinate of the top-left corner of the tile measured with
00567        respect to the reference grid. */
00568     uint_fast32_t tlx;
00569
00570     /* The y-coordinate of the top-left corner of the tile measured with
00571        respect to the reference grid. */
00572     uint_fast32_t tly;
00573
00574     /* The x-coordinate of the bottom-right corner of the tile measured
00575        with respect to the reference grid (plus one). */
00576     uint_fast32_t brx;
00577
00578     /* The y-coordinate of the bottom-right corner of the tile measured
00579        with respect to the reference grid (plus one). */
00580     uint_fast32_t bry;
00581
00582     /* The coding style. */
00583     uint_fast8_t csty;
00584
00585     /* The progression order. */
00586     uint_fast8_t prg;
00587
00588     /* The number of layers. */
00589     unsigned numlyrs;
00590
00591     /* The MCT to employ (if any). */
00592     uint_fast8_t mctid;
00593
00594     /* The packet iterator (used to determine the order of packet
00595        generation). */
00596     jpc_pi_t *pi;
00597
00598     /* The coding mode (i.e., integer or real). */
00599     bool intmode;
00600
00601     /* The number of bytes to allocate to the various layers. */
00602     uint_fast32_t *lyrsizes;
00603
00604     /* The number of tile-components. */
00605     unsigned numtcmts;
00606
00607     /* The per tile-component information. */
00608     jpc_enc_tcmt_t *tcmts;
00609
00610     /* The raw (i.e., uncompressed) size of this tile. */
00611     uint_fast32_t rawsize;
00612
00613 } jpc_enc_tile_t;
00614
00615 /* Encoder class. */
00616
00617 typedef struct jpc_enc_s {
00618
00619     /* The image being encoded. */
00620     jas_image_t *image;
00621
00622     /* The output stream. */
00623     jas_stream_t *out;
00624
00625     /* The coding parameters. */
00626     jpc_enc_cp_t *cp;
00627
00628     /* The tile currently being processed. */
00629     jpc_enc_tile_t *curtile;
00630
00631     /* The code stream state. */
00632     jpc_cstate_t *cstate;
00633
00634     /* The number of bytes output so far. */
00635     uint_fast32_t len;
00636
00637     /* The number of bytes available for the main body of the code stream. */
00638     /* This is used for rate allocation purposes. */
00639     uint_fast32_t mainbodysize;
00640
00641     /* The marker segment currently being processed. */
00642     /* This member is a convenience for making cleanup easier. */
00643     jpc_ms_t *mrk;
00644
00645     /* The stream used to temporarily hold tile-part data. */

```

```

00646         jas_stream_t *tmpstream;
00647
00648 } jpc_enc_t;
00649
00650 #endif

```

16.52 jpc_fix.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00064 /*
00065  * Fixed-Point Number Class
00066  *
00067  * $Id$
00068  */

```

```

00069
00070 #ifndef JPC_FIX_H
00071 #define JPC_FIX_H
00072
00073 /*****
00074  * Includes.
00075  *****/
00076
00077 #include "jasper/jas_config.h"
00078 #include "jasper/jas_types.h"
00079 #include "jasper/jas_fix.h"
00080 #include "jasper/jas_math.h"
00081
00082 /*****
00083  * Basic parameters of the fixed-point type.
00084  *****/
00085
00086 /*
00087  jpc_fix_t.
00088  The integral type used to represent a fixed-point number. This
00089  type must be capable of representing values from  $-(2^{31})$  to  $2^{31}-1$ 
00090  (inclusive).
00091
00092  jpc_fix_big_t.
00093  The integral type used to represent higher-precision intermediate results.
00094  This type should be capable of representing values from  $-(2^{63})$  to  $2^{63}-1$ 
00095  (inclusive).
00096
00097  JPC_FIX_FRACBITS.
00098  The number of bits used for the fractional part of a fixed-point number.
00099  */
00100
00101 typedef jas_fix_t jpc_fix_t;
00102 typedef jas_fix_big_t jpc_fix_big_t;
00103
00104 #if defined(JAS_ENABLE_32BIT)
00105 #define JPC_FIX_FRACBITS      13
00106 #else
00107 #define JPC_FIX_FRACBITS      18
00108 #endif
00109
00110 /*****
00111  * Instantiations of the generic fixed-point number macros for the
00112  * parameters given above. (Too bad C does not support templates, eh?)
00113  * The purpose of these macros is self-evident if one examines the
00114  * corresponding macros in the jasper/jas_fix.h header file.
00115  *****/
00116
00117 #define JPC_FIX_ZERO      JAS_FIX_ZERO(jpc_fix_t, JPC_FIX_FRACBITS)
00118 #define JPC_FIX_ONE       JAS_FIX_ONE(jpc_fix_t, JPC_FIX_FRACBITS)
00119 #define JPC_FIX_HALF      JAS_FIX_HALF(jpc_fix_t, JPC_FIX_FRACBITS)
00120
00121 JAS_ATTRIBUTE_CONST
00122 static inline jpc_fix_t jpc_inttofix(int x)
00123 {
00124     return JAS_INTTOFIX(jpc_fix_t, JPC_FIX_FRACBITS, x);
00125 }
00126
00127 JAS_ATTRIBUTE_CONST
00128 static inline int jpc_fixtoint(jpc_fix_t x)
00129 {
00130     return JAS_FIXTOINT(jpc_fix_t, JPC_FIX_FRACBITS, x);
00131 }
00132
00133 JAS_ATTRIBUTE_CONST
00134 static inline double jpc_fixtodbl(jpc_fix_t x)
00135 {
00136     return JAS_FIXTODBL(jpc_fix_t, JPC_FIX_FRACBITS, x);
00137 }
00138
00139 JAS_ATTRIBUTE_CONST
00140 static inline jpc_fix_t jpc_dbltofix(double x)
00141 {
00142     return JAS_DBLTOFIX(jpc_fix_t, JPC_FIX_FRACBITS, x);
00143 }
00144
00145 JAS_ATTRIBUTE_CONST
00146 static inline jpc_fix_t jpc_fix_add(jpc_fix_t x, jpc_fix_t y)
00147 {
00148     return JAS_FIX_ADD(jpc_fix_t, JPC_FIX_FRACBITS, x, y);
00149 }

```

```

00150
00151 JAS_ATTRIBUTE_CONST
00152 static inline jpc_fix_t jpc_fix_sub(jpc_fix_t x, jpc_fix_t y)
00153 {
00154     return JAS_FIX_SUB(jpc_fix_t, JPC_FIX_FRACBITS, x, y);
00155 }
00156
00157 JAS_ATTRIBUTE_CONST
00158 static inline jpc_fix_t jpc_fix_mul(jpc_fix_big_t x, jpc_fix_big_t y)
00159 {
00160     return JAS_FIX_MUL(jpc_fix_t, JPC_FIX_FRACBITS, jpc_fix_big_t, x, y);
00161 }
00162
00163 JAS_ATTRIBUTE_CONST
00164 static inline jpc_fix_big_t jpc_fix_mulbyint(jpc_fix_big_t x, int y)
00165 {
00166     return JAS_FIX_MUL(jpc_fix_t, JPC_FIX_FRACBITS, jpc_fix_big_t, x, y);
00167 }
00168
00169 JAS_ATTRIBUTE_CONST
00170 static inline jpc_fix_t jpc_fix_div(jpc_fix_big_t x, jpc_fix_t y)
00171 {
00172     return JAS_FIX_DIV(jpc_fix_t, JPC_FIX_FRACBITS, jpc_fix_big_t, x, y);
00173 }
00174
00175 JAS_ATTRIBUTE_CONST
00176 static inline jpc_fix_t jpc_fix_neg(jpc_fix_t x)
00177 {
00178     return JAS_FIX_NEG(jpc_fix_t, JPC_FIX_FRACBITS, x);
00179 }
00180
00181 //define      jpc_fix_asl(x, n)      JAS_FIX_ASL(jpc_fix_t, JPC_FIX_FRACBITS, x, n)
00182 //define      jpc_fix_asr(x, n)      JAS_FIX_ASR(jpc_fix_t, JPC_FIX_FRACBITS, x, n)
00183
00184 #if 0
00185 #ifndef JAS_ENABLE_32BIT
00186 #define jpc_fix_asl jas_least32_asl
00187 #define jpc_fix_asr jas_least32_asr
00188 #else
00189 #define jpc_fix_asl jas_fast32_asl
00190 #define jpc_fix_asr jas_fast32_asr
00191 #endif
00192 #endif
00193 #define jpc_fix_asl jas_fix_asl
00194 #define jpc_fix_asr jas_fix_asr
00195
00196 #define jpc_fix_pluseq(x, y) \
00197     JAS_FIX_PLUSEQ(jpc_fix_t, JPC_FIX_FRACBITS, x, y)
00198 #define jpc_fix_minuseq(x, y) \
00199     JAS_FIX_MINUSEQ(jpc_fix_t, JPC_FIX_FRACBITS, x, y)
00200 #define jpc_fix_muleq(x, y) \
00201     JAS_FIX_MULEQ(jpc_fix_t, JPC_FIX_FRACBITS, jpc_fix_big_t, x, y)
00202
00203 JAS_ATTRIBUTE_CONST
00204 static inline jpc_fix_t jpc_fix_abs(jpc_fix_t x)
00205 {
00206     return JAS_FIX_ABS(jpc_fix_t, JPC_FIX_FRACBITS, x);
00207 }
00208
00209 JAS_ATTRIBUTE_CONST
00210 static inline bool jpc_fix_isint(jpc_fix_t x)
00211 {
00212     return JAS_FIX_ISINT(jpc_fix_t, JPC_FIX_FRACBITS, x);
00213 }
00214
00215 JAS_ATTRIBUTE_CONST
00216 static inline int jpc_fix_sgn(jpc_fix_t x)
00217 {
00218     return JAS_FIX_SGN(jpc_fix_t, JPC_FIX_FRACBITS, x);
00219 }
00220
00221 JAS_ATTRIBUTE_CONST
00222 static inline jpc_fix_t jpc_fix_round(jpc_fix_t x)
00223 {
00224     return JAS_FIX_ROUND(jpc_fix_t, JPC_FIX_FRACBITS, x);
00225 }
00226
00227 JAS_ATTRIBUTE_CONST
00228 static inline jpc_fix_t jpc_fix_floor(jpc_fix_t x)
00229 {
00230     return JAS_FIX_FLOOR(jpc_fix_t, JPC_FIX_FRACBITS, x);

```

```

00231 }
00232
00233 /*****
00234  * Extra macros for convenience.
00235  *****/
00236
00237 /* Compute the sum of three fixed-point numbers. */
00238 JAS_ATTRIBUTE_CONST
00239 static inline jpc_fix_t jpc_fix_add3(jpc_fix_t x, jpc_fix_t y, jpc_fix_t z)
00240 {
00241     return jpc_fix_add(jpc_fix_add(x, y), z);
00242 }
00243
00244 #endif

```

16.53 jpc_flt.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.

```



```

00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00064 /*
00065  * Floating-Point Class
00066  *
00067  * $Id$
00068  */
00069
00070 #ifndef JPC_FLT_H
00071 #define JPC_FLT_H
00072
00073 /* The code ought to be modified so this type is not used at all. */
00074 /* Very few places in the code rely on floating-point arithmetic, aside
00075    from conversions in printf's. */
00076 typedef double jpc_flt_t;
00077
00078 #endif

```

16.54 jpc_math.h

```

00001 /*
00002  * Copyright (c) 2001-2002 Michael David Adams.
00003  * All rights reserved.
00004  */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007  *
00008  * JasPer License Version 2.0
00009  *
00010  * Copyright (c) 2001-2006 Michael David Adams
00011  * Copyright (c) 1999-2000 Image Power, Inc.
00012  * Copyright (c) 1999-2000 The University of British Columbia
00013  *
00014  * All rights reserved.
00015  *
00016  * Permission is hereby granted, free of charge, to any person (the
00017  * "User") obtaining a copy of this software and associated documentation
00018  * files (the "Software"), to deal in the Software without restriction,
00019  * including without limitation the rights to use, copy, modify, merge,
00020  * publish, distribute, and/or sell copies of the Software, and to permit
00021  * persons to whom the Software is furnished to do so, subject to the
00022  * following conditions:
00023  *
00024  * 1. The above copyright notices and this permission notice (which
00025  * includes the disclaimer below) shall be included in all copies or
00026  * substantial portions of the Software.
00027  *
00028  * 2. The name of a copyright holder shall not be used to endorse or
00029  * promote products derived from the Software without specific prior
00030  * written permission.
00031  *
00032  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00054  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,

```

```

00055  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058  *
00059  * __END_OF_JASPER_LICENSE__
00060  */
00061
00062 #ifndef JPC_MATH_H
00063 #define JPC_MATH_H
00064
00065 /*****
00066  * Includes
00067  *****/
00068
00069 #include "jasper/jas_config.h"
00070
00071 #include "jpc_fix.h"
00072
00073 /*****
00074  * Macros
00075  *****/
00076
00077 /* Compute the floor of the quotient of two integers. */
00078 #define JPC_FLOORDIV(x, y)      ((x) / (y))
00079
00080 /* Compute the ceiling of the quotient of two integers. */
00081 #define JPC_CEILDIV(x, y)      (((x) + (y) - 1) / (y))
00082
00083 /* Compute the floor of (x / 2^y). */
00084 #define JPC_FLOORDIVPOW2(x, y) ((x) >> (y))
00085
00086 /* Compute the ceiling of (x / 2^y). */
00087 #define JPC_CEILDIVPOW2(x, y)  (((x) + (1 << (y)) - 1) >> (y))
00088
00089 /*****
00090  * Functions.
00091  *****/
00092
00093 /* Calculate the bit position of the first leading one in a nonnegative
00094  integer. */
00095 JAS_ATTRIBUTE_CONST
00096 int jpc_int_firststone(int x);
00097
00098 JAS_ATTRIBUTE_CONST
00099 int jpc_fix_firststone(jpc_fix_t x);
00100
00101 /* Calculate the integer quantity floor(log2(x)), where x is a positive
00102  integer. */
00103 JAS_ATTRIBUTE_CONST
00104 unsigned jpc_floorlog2(uint_fast32_t x);
00105
00106 #endif

```

16.55 jpc_mct.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,

```

```

00022 * publish, distribute, and/or sell copies of the Software, and to permit
00023 * persons to whom the Software is furnished to do so, subject to the
00024 * following conditions:
00025 *
00026 * 1. The above copyright notices and this permission notice (which
00027 * includes the disclaimer below) shall be included in all copies or
00028 * substantial portions of the Software.
00029 *
00030 * 2. The name of a copyright holder shall not be used to endorse or
00031 * promote products derived from the Software without specific prior
00032 * written permission.
00033 *
00034 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060 *
00061 * __END_OF_JASPER_LICENSE__
00062 */
00063
00064 /*
00065 * Multicomponent Transform Code
00066 *
00067 * $Id$
00068 */
00069
00070 #ifndef JPC_MCT_H
00071 #define JPC_MCT_H
00072
00073 /*****
00074 * Includes.
00075 *****/
00076
00077 #include "jpc_fix.h"
00078
00079 #include "jasper/jas_seq.h"
00080
00081 /*****
00082 * Constants.
00083 *****/
00084
00085 /*
00086 * Multicomponent transform IDs.
00087 */
00088
00089 #define JPC_MCT_NONE 0
00090 #define JPC_MCT_ICT 1
00091 #define JPC_MCT_RCT 2
00092
00093 /*****
00094 * Functions.
00095 *****/
00096
00097 /* Calculate the forward RCT. */
00098 void jpc_rct(jas_matrix_t *c0, jas_matrix_t *c1, jas_matrix_t *c2);
00099
00100 /* Calculate the inverse RCT. */
00101 void jpc_irt(jas_matrix_t *c0, jas_matrix_t *c1, jas_matrix_t *c2);
00102

```

```

00103 /* Calculate the forward ICT. */
00104 void jpc_ict(jas_matrix_t *c0, jas_matrix_t *c1, jas_matrix_t *c2);
00105
00106 /* Calculate the inverse ICT. */
00107 void jpc_iict(jas_matrix_t *c0, jas_matrix_t *c1, jas_matrix_t *c2);
00108
00109 #endif

```

16.56 jpc_mqcod.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2003 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00064 /*
00065  * MQ Arithmetic Coder
00066  *

```

```

00067  * $Id$
00068  */
00069
00070 #ifndef JPC_MQCOD_H
00071 #define JPC_MQCOD_H
00072
00073 /*****
00074  * Includes.
00075  *****/
00076
00077 #include "jasper/jas_types.h"
00078
00079 /*****
00080  * Types.
00081  *****/
00082
00083 /*
00084  * MQ coder context information.
00085  */
00086
00087 typedef struct {
00088     /* The most probable symbol (MPS). */
00089     bool mps;
00090
00091     /* The state index. */
00092     int_least8_t ind;
00093 } jpc_mqctx_t;
00094
00095 /*
00096  * MQ coder state table entry.
00097  */
00098
00099 typedef struct jpc_mqstate_s {
00100     /* The Qe value. */
00101     uint_least16_t qeval;
00102
00103     /* The MPS. */
00104     bool mps;
00105
00106     /* The NMPS state. */
00107     const struct jpc_mqstate_s *nmpps;
00108
00109     /* The NLPS state. */
00110     const struct jpc_mqstate_s *nlps;
00111 } jpc_mqstate_t;
00112
00113 /*****
00114  * Data.
00115  *****/
00116
00117 /* The state table for the MQ coder. */
00118 extern const jpc_mqstate_t jpc_mqstates[];
00119
00120 #endif

```

16.57 jpc_mqdec.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2003 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  */

```

```

00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00064 /*
00065  * MQ Arithmetic Decoder
00066  *
00067  * $Id$
00068  */
00069
00070 #ifndef JPC_MQDEC_H
00071 #define JPC_MQDEC_H
00072
00073 /*****\
00074  * Includes.
00075  \*****/
00076
00077 #include "jasper/jas_types.h"
00078 #include "jasper/jas_stream.h"
00079
00080 #include "jpc_mqcdec.h"
00081
00082 #include <stdio.h>
00083
00084 /*****\
00085  * Types.
00086  \*****/
00087
00088 /* MQ arithmetic decoder. */
00089
00090 typedef struct {
00091
00092     /* The C register. */
00093     uint_least32_t creg;
00094
00095     /* The A register. */
00096     uint_least32_t areg;

```

```

00097
00098     /* The CT register. */
00099     uint_least32_t ctreg;
00100
00101     /* The current context. */
00102     const jpc_mqstate_t **curctx;
00103
00104     /* The per-context information. */
00105     const jpc_mqstate_t **ctxs;
00106
00107     /* The maximum number of contexts. */
00108     unsigned maxctxs;
00109
00110     /* The stream from which to read data. */
00111     jas_stream_t *in;
00112
00113     /* The last character read. */
00114     jas_uchar inbuffer;
00115
00116     /* The EOF indicator. */
00117     bool eof;
00118 } jpc_mqdec_t;
00119
00120
00121 /*****
00122  * Functions/macros for construction and destruction.
00123  *****/
00124
00125 /* Create a MQ decoder. */
00126 jpc_mqdec_t *jpc_mqdec_create(unsigned maxctxs, jas_stream_t *in);
00127
00128 /* Destroy a MQ decoder. */
00129 void jpc_mqdec_destroy(jpc_mqdec_t *dec);
00130
00131 /*****
00132  * Functions/macros for initialization.
00133  *****/
00134
00135 /* Set the input stream associated with a MQ decoder. */
00136 void jpc_mqdec_setinput(jpc_mqdec_t *dec, jas_stream_t *in);
00137
00138 /* Initialize a MQ decoder. */
00139 void jpc_mqdec_init(jpc_mqdec_t *dec);
00140
00141 /*****
00142  * Functions/macros for manipulating contexts.
00143  *****/
00144
00145 /* Set the current context for a MQ decoder. */
00146 static inline void jpc_mqdec_setcurctx(jpc_mqdec_t *dec, unsigned ctxno)
00147 {
00148     dec->curctx = &dec->ctxs[ctxno];
00149 }
00150
00151 /* Set the state information for all contexts of a MQ decoder. */
00152 void jpc_mqdec_setctxs(const jpc_mqdec_t *dec, unsigned numctxs, const jpc_mqctx_t *ctxs);
00153
00154 /*****
00155  * Functions/macros for decoding bits.
00156  *****/
00157
00158 /* Decode a symbol. */
00159 #ifdef NDEBUG
00160 #define jpc_mqdec_getbit(dec) \
00161     jpc_mqdec_getbit_macro(dec)
00162 #else
00163 #define jpc_mqdec_getbit(dec) \
00164     jpc_mqdec_getbit_func(dec)
00165 #endif
00166
00167 /* Decode a symbol (assuming an unskewed probability distribution). */
00168 #ifdef NDEBUG
00169 #define jpc_mqdec_getbitnoskew(dec) \
00170     jpc_mqdec_getbit_macro(dec)
00171 #else
00172 #define jpc_mqdec_getbitnoskew(dec) \
00173     jpc_mqdec_getbit_func(dec)
00174 #endif
00175
00176 /*****
00177  * Functions/macros for debugging.

```

```

00178 \*****/
00179
00180 /* Dump the MQ decoder state for debugging. */
00181 void jpc_mqdec_dump(const jpc_mqdec_t *dec);
00182
00183 \*****/
00184 * EVERYTHING BELOW THIS POINT IS IMPLEMENTATION SPECIFIC AND NOT PART OF THE
00185 * APPLICATION INTERFACE. DO NOT RELY ON ANY OF THE INTERNAL FUNCTIONS/MACROS
00186 * GIVEN BELOW.
00187 \*****/
00188
00189 bool jpc_mqdec_mpsexchrenormd(jpc_mqdec_t *dec);
00190 bool jpc_mqdec_lpsexchrenormd(jpc_mqdec_t *dec);
00191
00192 JAS_FORCE_INLINE
00193 static bool jpc_mqdec_getbit_macro(jpc_mqdec_t *dec)
00194 {
00195     const jpc_mqstate_t *const state = *dec->curctx;
00196
00197     dec->areg -= state->qeval;
00198
00199     if (dec->creg >= (uint_least32_t)state->qeval << 16) {
00200         dec->creg -= (uint_least32_t)state->qeval << 16;
00201         return dec->areg & 0x8000
00202             ? state->mps
00203             : jpc_mqdec_mpsexchrenormd(dec);
00204     } else {
00205         return jpc_mqdec_lpsexchrenormd(dec);
00206     }
00207 }
00208
00209 JAS_FORCE_INLINE
00210 static bool jpc_mqdec_mpsexchange(uint_least32_t areg, uint_least32_t delta, const jpc_mqstate_t **curctx)
00211 {
00212     if (areg < delta) {
00213         const jpc_mqstate_t *state = *curctx;
00214         /* LPS decoded. */
00215         *curctx = state->nmps;
00216         return !state->mps;
00217     } else {
00218         const jpc_mqstate_t *state = *curctx;
00219         /* MPS decoded. */
00220         *curctx = state->nmps;
00221         return state->mps;
00222     }
00223 }
00224
00225 JAS_FORCE_INLINE
00226 static bool jpc_mqdec_lpsexchange(uint_least32_t *areg_p, uint_least32_t delta, const jpc_mqstate_t
**curctx)
00227 {
00228     if (*areg_p >= delta) {
00229         const jpc_mqstate_t *state = *curctx;
00230         *areg_p = delta;
00231         *curctx = state->nmps;
00232         return !state->mps;
00233     } else {
00234         const jpc_mqstate_t *state = *curctx;
00235         *areg_p = delta;
00236         *curctx = state->nmps;
00237         return state->mps;
00238     }
00239 }
00240
00241 bool jpc_mqdec_getbit_func(jpc_mqdec_t *dec);
00242
00243 #endif

```

16.58 jpc_mqenc.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */

```



```

00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00064 /*
00065  * MQ Arithmetic Encoder
00066  *
00067  * $Id$
00068  */
00069
00070 #ifndef JPC_MQENC_H
00071 #define JPC_MQENC_H
00072
00073 /*****\
00074  * Includes.
00075  \*****/
00076
00077 #include "jasper/jas_types.h"
00078 #include "jasper/jas_stream.h"
00079
00080 #include "jpc_mqcod.h"
00081
00082 #include <stdio.h>
00083
00084 /*****\
00085  * Constants.
00086  \*****/
00087

```

```

00088 /*
00089  * Termination modes.
00090  */
00091
00092 #define JPC_MQENC_DEFTERM      0      /* default termination */
00093 #define JPC_MQENC_PTERM       1      /* predictable termination */
00094
00095 /*****
00096  * Types.
00097  \*****/
00098
00099 /* MQ arithmetic encoder class. */
00100
00101 typedef struct {
00102
00103     /* The C register. */
00104     uint_least32_t creg;
00105
00106     /* The A register. */
00107     uint_least32_t areg;
00108
00109     /* The CT register. */
00110     uint_least32_t ctreg;
00111
00112     /* The maximum number of contexts. */
00113     unsigned maxctxs;
00114
00115     /* The per-context information. */
00116     const jpc_mqstate_t **ctxs;
00117
00118     /* The current context. */
00119     const jpc_mqstate_t **curctx;
00120
00121     /* The stream for encoder output. */
00122     jas_stream_t *out;
00123
00124     /* The byte buffer (i.e., the B variable in the standard). */
00125     int_least16_t outbuf;
00126
00127     /* The last byte output. */
00128     int_least16_t lastbyte;
00129
00130     /* The error indicator. */
00131     bool err;
00132
00133 } jpc_mqenc_t;
00134
00135 /* MQ arithmetic encoder state information. */
00136
00137 typedef struct {
00138
00139     /* The A register. */
00140     unsigned areg;
00141
00142     /* The C register. */
00143     unsigned creg;
00144
00145     /* The CT register. */
00146     unsigned ctreg;
00147
00148     /* The last byte output by the encoder. */
00149     int lastbyte;
00150
00151 } jpc_mqencstate_t;
00152
00153 /*****
00154  * Functions/macros for construction and destruction.
00155  \*****/
00156
00157 /* Create a MQ encoder. */
00158 jpc_mqenc_t *jpc_mqenc_create(unsigned maxctxs, jas_stream_t *out);
00159
00160 /* Destroy a MQ encoder. */
00161 void jpc_mqenc_destroy(jpc_mqenc_t *enc);
00162
00163 /*****
00164  * Functions/macros for initialization.
00165  \*****/
00166
00167 /* Initialize a MQ encoder. */
00168 void jpc_mqenc_init(jpc_mqenc_t *enc);

```

```

00169
00170 /*****
00171  * Functions/macros for context manipulation.
00172  *****/
00173
00174 /* Set the current context. */
00175 static inline void jpc_mqenc_setcurctx(jpc_mqenc_t *enc, unsigned ctxno) {
00176     enc->curctx = &enc->ctxs[ctxno];
00177 }
00178
00179 /* Set the state information for multiple contexts. */
00180 void jpc_mqenc_setctxs(jpc_mqenc_t *enc, unsigned numctxs, const jpc_mqctx_t *ctxs);
00181
00182 /*****
00183  * Miscellaneous functions/macros.
00184  *****/
00185
00186 /* Get the error state of a MQ encoder. */
00187 static inline bool jpc_mqenc_error(const jpc_mqenc_t *enc) {
00188     return enc->err;
00189 }
00190
00191 /* Get the current encoder state. */
00192 void jpc_mqenc_getstate(const jpc_mqenc_t *enc, jpc_mqencstate_t *state);
00193
00194 /* Terminate the code. */
00195 int jpc_mqenc_flush(jpc_mqenc_t *enc, int termmode);
00196
00197 /*****
00198  * Functions/macros for encoding bits.
00199  *****/
00200
00201 /*****
00202  * Functions/macros for debugging.
00203  *****/
00204
00205 int jpc_mqenc_dump(const jpc_mqenc_t *mqenc);
00206
00207 /*****
00208  * Implementation-specific details.
00209  *****/
00210
00211 /* Note: These function prototypes are included only to satisfy the
00212  needs of the mqenc_putbit_macro macro. Do not call any of these
00213  functions directly. */
00214 int jpc_mqenc_codemps2(jpc_mqenc_t *enc);
00215 int jpc_mqenc_codelps(jpc_mqenc_t *enc);
00216
00217 int jpc_mqenc_putbit(jpc_mqenc_t *enc, bool bit);
00218
00219 #endif

```

16.59 jpc_qmfb.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2004 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit

```

```

00023 * persons to whom the Software is furnished to do so, subject to the
00024 * following conditions:
00025 *
00026 * 1. The above copyright notices and this permission notice (which
00027 * includes the disclaimer below) shall be included in all copies or
00028 * substantial portions of the Software.
00029 *
00030 * 2. The name of a copyright holder shall not be used to endorse or
00031 * promote products derived from the Software without specific prior
00032 * written permission.
00033 *
00034 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060 *
00061 * __END_OF_JASPER_LICENSE__
00062 */
00063
00064 /*
00065 * Quadrature Mirror-Image Filter Bank (QMFB) Routines
00066 *
00067 * $Id$
00068 */
00069
00070 #ifndef JPC_QMFB_H
00071 #define JPC_QMFB_H
00072
00073 /*****\
00074 * Includes.
00075 \*****/
00076
00077 #include "jpc_fix.h"
00078
00079 /*****\
00080 * Constants.
00081 \*****/
00082
00083 /* QMFB IDs. */
00084 #define JPC_QMFB1D_FT 1 /* 5/3 */
00085 #define JPC_QMFB1D_NS 2 /* 9/7 */
00086
00087 /*****\
00088 * Types.
00089 \*****/
00090
00091 /*****\
00092 * Functions.
00093 \*****/
00094
00095 #if !defined(JPC_QMFB_COLGRPSIZE)
00096 /* The number of columns to group together during the vertical processing
00097 stage of the wavelet transform. */
00098 /* The default value for this parameter is probably not optimal for
00099 any particular platform. Hopefully, it is not too unreasonable, however. */
00100 #define JPC_QMFB_COLGRPSIZE 16
00101 #endif
00102
00103 typedef struct {

```

```

00104         int (*analyze)(jpc_fix_t *, int, int, int, int, int);
00105         int (*synthesize)(jpc_fix_t *, int, int, int, int, int);
00106         const double *lpenergywts;
00107         const double *hpenergywts;
00108     } jpc_qmfb2d_t;
00109
00110 extern const jpc_qmfb2d_t jpc_ft_qmfb2d;
00111 extern const jpc_qmfb2d_t jpc_ns_qmfb2d;
00112
00113 #endif

```

16.60 jpc_t1cod.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063

```

```

00064 /*
00065  * $Id$
00066 */
00067
00068 #ifndef JPC_T1COD_H
00069 #define JPC_T1COD_H
00070
00071 /*****
00072  * Includes.
00073  *****/
00074
00075 #include "jpc_fix.h"
00076 #include "jpc_mqcod.h"
00077 #include "jpc_tsfb.h"
00078 #include "jasper/jas_math.h"
00079
00080 /*****
00081  * Constants.
00082  *****/
00083
00084 /* The number of bits used to index into various lookup tables. */
00085 #define JPC_NMSEDEC_BITS 7
00086 #define JPC_NMSEDEC_FRACBITS (JPC_NMSEDEC_BITS - 1)
00087
00088 /*
00089  * Segment types.
00090  */
00091
00092 enum jpc_segtype {
00093     JPC_SEG_INVALID,
00094     /* MQ. */
00095     JPC_SEG_MQ,
00096     /* Raw. */
00097     JPC_SEG_RAW,
00098 };
00099
00100 /* The nominal word size. */
00101 #define JPC_PREC 32
00102
00103 /* Tier-1 coding pass types. */
00104 enum jpc_passtype {
00105     JPC_SIGPASS, /*< significance */
00106     JPC_REFPASS, /*< refinement */
00107     JPC_CLNPASS, /*< cleanup */
00108 };
00109
00110 /*
00111  * Per-sample state information for tier-1 coding.
00112  */
00113
00114 /* The northeast neighbour has been found to be significant. */
00115 #define JPC_NESIG 0x0001
00116 /* The southeast neighbour has been found to be significant. */
00117 #define JPC_SESIG 0x0002
00118 /* The southwest neighbour has been found to be significant. */
00119 #define JPC_SWSIG 0x0004
00120 /* The northwest neighbour has been found to be significant. */
00121 #define JPC_NWSIG 0x0008
00122 /* The north neighbour has been found to be significant. */
00123 #define JPC_NSIG 0x0010
00124 /* The east neighbour has been found to be significant. */
00125 #define JPC_ESIG 0x0020
00126 /* The south neighbour has been found to be significant. */
00127 #define JPC_SSIG 0x0040
00128 /* The west neighbour has been found to be significant. */
00129 #define JPC_WSIG 0x0080
00130 /* The significance mask for 8-connected neighbours. */
00131 #define JPC_OTHSIGMSK \
00132     (JPC_NSIG | JPC_NESIG | JPC_ESIG | JPC_SESIG | JPC_SSIG | JPC_SWSIG | JPC_WSIG | JPC_NWSIG)
00133 /* The significance mask for 4-connected neighbours. */
00134 #define JPC_PRMSIGMSK (JPC_NSIG | JPC_ESIG | JPC_SSIG | JPC_WSIG)
00135
00136 /* The north neighbour is negative in value. */
00137 #define JPC_NSGN 0x0100
00138 /* The east neighbour is negative in value. */
00139 #define JPC_ESGN 0x0200
00140 /* The south neighbour is negative in value. */
00141 #define JPC_SSGN 0x0400
00142 /* The west neighbour is negative in value. */

```

```

00146 #define JPC_WSGN          0x0800
00147 /* The sign mask for 4-connected neighbours. */
00148 #define JPC_SGNMSK        (JPC_NSNG | JPC_ESNG | JPC_SSGN | JPC_WSGN)
00149
00150 /* This sample has been found to be significant. */
00151 #define JPC_SIG           0x1000
00152 /* The sample has been refined. */
00153 #define JPC_REFINE        0x2000
00154 /* This sample has been processed during the significance pass. */
00155 #define JPC_VISIT         0x4000
00156
00157 /* The number of aggregation contexts. */
00158 #define JPC_NUMAGGCTXS    1
00159 /* The number of zero coding contexts. */
00160 #define JPC_NUMZCCTXS     9
00161 /* The number of magnitude contexts. */
00162 #define JPC_NUMMAGCTXS    3
00163 /* The number of sign coding contexts. */
00164 #define JPC_NUMSCCTXS     5
00165 /* The number of uniform contexts. */
00166 #define JPC_NUMUCTXS      1
00167
00168 /* The context ID for the first aggregation context. */
00169 #define JPC_AGGCTXNO      0
00170 /* The context ID for the first zero coding context. */
00171 #define JPC_ZCCTXNO       (JPC_AGGCTXNO + JPC_NUMAGGCTXS)
00172 /* The context ID for the first magnitude context. */
00173 #define JPC_MAGCTXNO      (JPC_ZCCTXNO + JPC_NUMZCCTXS)
00174 /* The context ID for the first sign coding context. */
00175 #define JPC_SCCTXNO       (JPC_MAGCTXNO + JPC_NUMMAGCTXS)
00176 /* The context ID for the first uniform context. */
00177 #define JPC_UCTXNO        (JPC_SCCTXNO + JPC_NUMSCCTXS)
00178 /* The total number of contexts. */
00179 #define JPC_NUMCTXS       (JPC_UCTXNO + JPC_NUMUCTXS)
00180
00181 /*****
00182  * External data.
00183  *****/
00184
00185 /* These lookup tables are used by various macros/functions. */
00186 /* Do not access these lookup tables directly. */
00187 extern uint_least8_t jpc_zcctxnolut[];
00188 extern bool jpc_spblut[];
00189 extern uint_least8_t jpc_scctxnolut[];
00190 extern uint_least8_t jpc_magctxnolut[];
00191 extern jpc_fix_t jpc_refnmsedec[];
00192 extern jpc_fix_t jpc_signmsedec[];
00193 extern jpc_fix_t jpc_refnmsedec0[];
00194 extern jpc_fix_t jpc_signmsedec0[];
00195
00196 /* The initial settings for the MQ contexts. */
00197 extern jpc_mqctx_t jpc_mqctxs[];
00198
00199 /*****
00200  * Functions and macros.
00201  *****/
00202
00203 /* Arithmetic shift right (with ability to shift left also). */
00204 JAS_ATTRIBUTE_CONST
00205 static inline jpc_fix_t JPC_ASR(jpc_fix_t x, int n)
00206 {
00207     return n >= 0
00208         ? x » n
00209         : x « -n;
00210 }
00211
00212 /* Get the zero coding context. */
00213 JAS_ATTRIBUTE_CONST
00214 static inline uint_least8_t JPC_GETZCCTXNO(unsigned f, enum jpc_tsfb_orient orient)
00215 {
00216     return jpc_zcctxnolut[((unsigned)orient « 8) | (f & JPC_OTHSIGMSK)];
00217 }
00218
00219 /* Get the sign prediction bit. */
00220 JAS_ATTRIBUTE_CONST
00221 static inline bool JPC_GETSPB(unsigned f)
00222 {
00223     return jpc_spblut[(f & (JPC_PRIMSIGMSK | JPC_SGNMSK)) » 4];
00224 }
00225
00226 /* Get the sign coding context. */

```

```

00227 JAS_ATTRIBUTE_CONST
00228 static inline uint_least8_t JPC_GETSCCTXNO(unsigned f)
00229 {
00230     return jpc_scctxnolut[(f & (JPC_PRIMSIGMSK | JPC_SGNMSK)) >> 4];
00231 }
00232
00233 /* Get the magnitude context. */
00234 JAS_ATTRIBUTE_CONST
00235 static inline uint_least8_t JPC_GETMAGCTXNO(unsigned f)
00236 {
00237     return jpc_magctxnolut[(f & JPC_OTHSIGMSK) | (((f & JPC_REFINE) != 0) << 11)];
00238 }
00239
00240 /* Get the normalized MSE reduction for significance passes. */
00241 JAS_ATTRIBUTE_CONST
00242 static inline jpc_fix_t JPC_GETSIGNMSEDEC(jpc_fix_t x, int bitpos)
00243 {
00244     return bitpos > JPC_NMSEDEC_FRACBITS
00245         ? jpc_signmsedec[JPC_ASR(x, bitpos - JPC_NMSEDEC_FRACBITS) & JAS_ONES(JPC_NMSEDEC_BITS)]
00246         : jpc_signmsedec0[JPC_ASR(x, bitpos - JPC_NMSEDEC_FRACBITS) & JAS_ONES(JPC_NMSEDEC_BITS)];
00247 }
00248
00249 /* Get the normalized MSE reduction for refinement passes. */
00250 JAS_ATTRIBUTE_CONST
00251 static inline jpc_fix_t JPC_GETREFNMSEDEC(jpc_fix_t x, int bitpos)
00252 {
00253     return bitpos > JPC_NMSEDEC_FRACBITS
00254         ? jpc_refnmsedec[JPC_ASR(x, bitpos - JPC_NMSEDEC_FRACBITS) & JAS_ONES(JPC_NMSEDEC_BITS)]
00255         : jpc_refnmsedec0[JPC_ASR(x, bitpos - JPC_NMSEDEC_FRACBITS) & JAS_ONES(JPC_NMSEDEC_BITS)];
00256 }
00257
00258 /* Update the per-sample state information. */
00259 static inline void JPC_UPDATEFLAGS4(jpc_fix_t *fp, unsigned rowstep, bool s, bool vcausalflag)
00260 {
00261     jpc_fix_t *np = fp - rowstep;
00262     jpc_fix_t *sp = fp + rowstep;
00263     if (vcausalflag) {
00264         sp[-1] |= JPC_NESIG;
00265         sp[1] |= JPC_NWSIG;
00266         if (s) {
00267             *sp |= JPC_NSIG | JPC_NSGN;
00268             fp[-1] |= JPC_ESIG | JPC_ESGN;
00269             fp[1] |= JPC_WSIG | JPC_WSGN;
00270         } else {
00271             *sp |= JPC_NSIG;
00272             fp[-1] |= JPC_ESIG;
00273             fp[1] |= JPC_WSIG;
00274         }
00275     } else {
00276         np[-1] |= JPC_SESIG;
00277         np[1] |= JPC_SWSIG;
00278         sp[-1] |= JPC_NESIG;
00279         sp[1] |= JPC_NWSIG;
00280         if (s) {
00281             *np |= JPC_SSIG | JPC_SSGN;
00282             *sp |= JPC_NSIG | JPC_NSGN;
00283             fp[-1] |= JPC_ESIG | JPC_ESGN;
00284             fp[1] |= JPC_WSIG | JPC_WSGN;
00285         } else {
00286             *np |= JPC_SSIG;
00287             *sp |= JPC_NSIG;
00288             fp[-1] |= JPC_ESIG;
00289             fp[1] |= JPC_WSIG;
00290         }
00291     }
00292 }
00293
00294 /* Initialize the lookup tables used by the codec. */
00295 void jpc_initluts(void);
00296
00297 void jpc_initmqctxs(void);
00298
00299 /* Get the nominal gain associated with a particular band. */
00300 JAS_ATTRIBUTE_CONST
00301 unsigned JPC_NOMINALGAIN(unsigned qmfbid, unsigned numlvls, unsigned lvlno, enum jpc_tsfb_orient orient);
00302
00303 /* Get the coding pass type. */
00304 JAS_ATTRIBUTE_CONST
00305 enum jpc_passtype JPC_PASSTYPE(unsigned passno);
00306
00307 /* Get the segment type. */

```



```

00308 JAS_ATTRIBUTE_CONST
00309 enum jpc_segtype JPC_SEGTYPE(unsigned passno, unsigned firstpassno, bool bypass);
00310
00311 /* Get the number of coding passess in the segment. */
00312 JAS_ATTRIBUTE_CONST
00313 unsigned JPC_SEGPASSCNT(unsigned passno, unsigned firstpassno, unsigned numpasses, bool bypass,
00314     bool termall);
00315
00316 /* Is the coding pass terminated? */
00317 JAS_ATTRIBUTE_CONST
00318 bool JPC_ISTERMINATED(unsigned passno, unsigned firstpassno, unsigned numpasses, bool termall,
00319     bool lazy);
00320
00321 #endif

```

16.61 jpc_t1dec.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.

```

```

00060 *
00061 * __END_OF_JASPER_LICENSE__
00062 */
00063
00064 /*
00065 * Tier 1 Decoder
00066 *
00067 * $Id$
00068 */
00069
00070 #ifndef JPC_T1DEC_H
00071 #define JPC_T1DEC_H
00072
00073 /*****\
00074 * Includes.
00075 \*****/
00076
00077 #include "jpc_dec.h"
00078
00079 /*****\
00080 * Functions.
00081 \*****/
00082
00083 /* Decode all of the code blocks for a particular tile. */
00084 int jpc_dec_decodeblks(jpc_dec_t *dec, jpc_dec_tile_t *tile);
00085
00086 #endif

```

16.62 jpc_t1enc.h

```

00001 /*
00002 * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003 * British Columbia.
00004 * Copyright (c) 2001-2002 Michael David Adams.
00005 * All rights reserved.
00006 */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009 *
00010 * JasPer License Version 2.0
00011 *
00012 * Copyright (c) 2001-2006 Michael David Adams
00013 * Copyright (c) 1999-2000 Image Power, Inc.
00014 * Copyright (c) 1999-2000 The University of British Columbia
00015 *
00016 * All rights reserved.
00017 *
00018 * Permission is hereby granted, free of charge, to any person (the
00019 * "User") obtaining a copy of this software and associated documentation
00020 * files (the "Software"), to deal in the Software without restriction,
00021 * including without limitation the rights to use, copy, modify, merge,
00022 * publish, distribute, and/or sell copies of the Software, and to permit
00023 * persons to whom the Software is furnished to do so, subject to the
00024 * following conditions:
00025 *
00026 * 1. The above copyright notices and this permission notice (which
00027 * includes the disclaimer below) shall be included in all copies or
00028 * substantial portions of the Software.
00029 *
00030 * 2. The name of a copyright holder shall not be used to endorse or
00031 * promote products derived from the Software without specific prior
00032 * written permission.
00033 *
00034 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.

```

```

00047 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060 *
00061 * __END_OF_JASPER_LICENSE__
00062 */
00063
00064 /*
00065  * Tier 1 Encoder
00066  *
00067  * $Id$
00068  */
00069
00070 #ifndef JPC_T1ENC_H
00071 #define JPC_T1ENC_H
00072
00073 /*****\
00074  * Includes.
00075  \*****/
00076
00077 #include "jpc_enc.h"
00078
00079 /*****\
00080  * Functions.
00081  \*****/
00082
00083 /* Encode all of the code blocks. */
00084 int jpc_enc_encblks(jpc_enc_t *enc);
00085
00086 #endif

```

16.63 jpc_t2cod.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  */

```

```

00034 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060 *
00061 * __END_OF_JASPER_LICENSE__
00062 */
00063
00064 /*
00065  * Tier-2 Coding Library
00066  *
00067  * $Id$
00068  */
00069
00070 #ifndef JPC_T2COD_H
00071 #define JPC_T2COD_H
00072
00073 /*****\
00074  * Includes.
00075  \*****/
00076
00077 #include "jpc_cs.h"
00078
00079 /*****\
00080  * Types.
00081  \*****/
00082
00083 /* Progression change list. */
00084
00085 typedef struct {
00086
00087     /* The number of progression changes. */
00088     unsigned numpchgs;
00089
00090     /* The maximum number of progression changes that can be accomodated
00091        without growing the progression change array. */
00092     unsigned maxpchgs;
00093
00094     /* The progression changes. */
00095     jpc_pchg_t **pchgs;
00096
00097 } jpc_pchglst_t;
00098
00099 /* Packet iterator per-resolution-level information. */
00100
00101 typedef struct {
00102
00103     /* The number of precincts. */
00104     unsigned numprcs;
00105
00106     /* The last layer processed for each precinct. */
00107     unsigned *prclyrns;
00108
00109     /* The precinct width exponent. */
00110     unsigned prcwidthexpn;
00111
00112     /* The precinct height exponent. */
00113     unsigned prcheightexpn;
00114

```

```

00115         /* The number of precincts spanning the resolution level in the horizontal
00116         direction. */
00117         unsigned numhprcs;
00118     } jpc_pirlvl_t;
00119 } jpc_pirlvl_t;
00120
00121 /* Packet iterator per-component information. */
00122
00123 typedef struct {
00124     /* The number of resolution levels. */
00125     unsigned numrlvls;
00126
00127     /* The per-resolution-level information. */
00128     jpc_pirlvl_t *pirlvls;
00129
00130     /* The horizontal sampling period. */
00131     uint_fast32_t hsamp;
00132
00133     /* The vertical sampling period. */
00134     uint_fast32_t vsamp;
00135 } jpc_picomp_t;
00136
00137 } jpc_picomp_t;
00138
00139 /* Packet iterator class. */
00140
00141 typedef struct {
00142     /* The number of layers. */
00143     unsigned numlyrs;
00144
00145     /* The number of resolution levels. */
00146     unsigned maxrlvls;
00147
00148     /* The number of components. */
00149     unsigned numcomps;
00150
00151     /* The per-component information. */
00152     jpc_picomp_t *picomps;
00153
00154     /* The current component. */
00155     jpc_picomp_t *picomp;
00156
00157     /* The current resolution level. */
00158     jpc_pirlvl_t *pirlvl;
00159
00160     /* The number of the current component. */
00161     unsigned compno;
00162
00163     /* The number of the current resolution level. */
00164     unsigned rlvln;
00165
00166     /* The number of the current precinct. */
00167     unsigned prcno;
00168
00169     /* The number of the current layer. */
00170     unsigned lyrno;
00171
00172     /* The x-coordinate of the current position. */
00173     uint_fast32_t x;
00174
00175     /* The y-coordinate of the current position. */
00176     uint_fast32_t y;
00177
00178     /* The horizontal step size. */
00179     uint_fast32_t xstep;
00180
00181     /* The vertical step size. */
00182     uint_fast32_t ystep;
00183
00184     /* The x-coordinate of the top-left corner of the tile on the reference
00185     grid. */
00186     uint_fast32_t xstart;
00187
00188     /* The y-coordinate of the top-left corner of the tile on the reference
00189     grid. */
00190     uint_fast32_t ystart;
00191
00192     /* The x-coordinate of the bottom-right corner of the tile on the
00193     reference grid (plus one). */
00194     uint_fast32_t xend;
00195 }

```

```

00196
00197     /* The y-coordinate of the bottom-right corner of the tile on the
00198        reference grid (plus one). */
00199     uint_fast32_t yend;
00200
00201     /* The current progression change. */
00202     const jpc_pchg_t *pchg;
00203
00204     /* The progression change list. */
00205     jpc_pchglst_t *pchglst;
00206
00207     /* The progression to use in the absense of explicit specification. */
00208     jpc_pchg_t defaultpchg;
00209
00210     /* The current progression change number. */
00211     int pchgno;
00212
00213     /* Is this the first time in the current progression volume? */
00214     bool prgvolfirst;
00215
00216     /* Is the current iterator value valid? */
00217     bool valid;
00218
00219     /* The current packet number. */
00220     int pktno;
00221
00222 } jpc_pi_t;
00223
00224 /*****
00225  * Functions/macros for packet iterators.
00226  *****/
00227
00228 /* Create a packet iterator. */
00229 jpc_pi_t *jpc_pi_create0(void);
00230
00231 /* Destroy a packet iterator. */
00232 void jpc_pi_destroy(jpc_pi_t *pi);
00233
00234 /* Add a progression change to a packet iterator. */
00235 int jpc_pi_addpchg(jpc_pi_t *pi, jpc_pocpchg_t *pchg);
00236
00237 /* Prepare a packet iterator for iteration. */
00238 int jpc_pi_init(jpc_pi_t *pi);
00239
00240 /* Set the iterator to the first packet. */
00241 int jpc_pi_begin(jpc_pi_t *pi);
00242
00243 /* Proceed to the next packet in sequence. */
00244 int jpc_pi_next(jpc_pi_t *pi);
00245
00246 /* Get the index of the current packet. */
00247 #define jpc_pi_getind(pi)      ((pi)->pktno)
00248
00249 /* Get the component number of the current packet. */
00250 #define jpc_pi_cmpno(pi)      (assert(pi->valid), (pi)->compno)
00251
00252 /* Get the resolution level of the current packet. */
00253 #define jpc_pi_rlvlno(pi)     (assert(pi->valid), (pi)->rlvlno)
00254
00255 /* Get the layer number of the current packet. */
00256 #define jpc_pi_lyrno(pi)      (assert(pi->valid), (pi)->lyrno)
00257
00258 /* Get the precinct number of the current packet. */
00259 #define jpc_pi_prcno(pi)      (assert(pi->valid), (pi)->prcno)
00260
00261 /* Get the progression order for the current packet. */
00262 #define jpc_pi_prg(pi)        (assert(pi->valid), (pi)->pchg->prgord)
00263
00264 /*****
00265  * Functions/macros for progression change lists.
00266  *****/
00267
00268 /* Create a progression change list. */
00269 jpc_pchglst_t *jpc_pchglst_create(void);
00270
00271 /* Destroy a progression change list. */
00272 void jpc_pchglst_destroy(jpc_pchglst_t *pchglst);
00273
00274 /* Insert a new element into a progression change list. */
00275 int jpc_pchglst_insert(jpc_pchglst_t *pchglst, int pchgno, jpc_pchg_t *pchg);
00276

```

```

00277 /* Remove an element from a progression change list. */
00278 jpc_pchg_t *jpc_pchglst_remove(jpc_pchglst_t *pchglst, unsigned pchgno);
00279
00280 /* Get an element from a progression change list. */
00281 JAS_ATTRIBUTE_PURE
00282 const jpc_pchg_t *jpc_pchglst_get(const jpc_pchglst_t *pchglst, unsigned pchgno);
00283
00284 /* Copy a progression change list. */
00285 jpc_pchglst_t *jpc_pchglst_copy(const jpc_pchglst_t *pchglst);
00286
00287 /* Get the number of elements in a progression change list. */
00288 JAS_ATTRIBUTE_PURE
00289 unsigned jpc_pchglst_numpchgs(const jpc_pchglst_t *pchglst);
00290
00291 /*****
00292  * Functions/macros for progression changes.
00293  *****/
00294
00295 /* Destroy a progression change. */
00296 void jpc_pchg_destroy(jpc_pchg_t *pchg);
00297
00298 /* Copy a progression change. */
00299 jpc_pchg_t *jpc_pchg_copy(const jpc_pchg_t *pchg);
00300
00301 /*****
00302  * Functions/macros for debugging.
00303  *****/
00304
00305 void jpc_pi_dump(const jpc_pi_t *pi);
00306
00307 #endif

```

16.64 jpc_t2dec.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,

```

```

00043 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060 *
00061 * __END_OF_JASPER_LICENSE__
00062 */
00063
00064 /*
00065  * Tier 2 Decoder
00066  *
00067  * $Id$
00068  */
00069
00070 #ifndef JPC_T2DEC_H
00071 #define JPC_T2DEC_H
00072
00073 /*****
00074  * Includes.
00075  *****/
00076
00077 #include "jasper/jas_stream.h"
00078
00079 #include "jpc_dec.h"
00080 #include "jpc_t2cod.h"
00081
00082 /*****
00083  * Functions.
00084  *****/
00085
00086 /* Decode the packets for a tile-part. */
00087 int jpc_dec_decodepkts(jpc_dec_t *dec, jas_stream_t *pkthdrstream,
00088     jas_stream_t *in);
00089
00090 /* Create a packet iterator for the decoder. */
00091 jpc_pi_t *jpc_dec_pi_create(jpc_dec_t *dec, jpc_dec_tile_t *tile);
00092
00093 #endif

```

16.65 jpc_t2enc.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit

```



```

00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00064 /*
00065  * Tier 2 Encoder
00066  *
00067  * $Id$
00068  */
00069
00070 #ifndef JPC_T2ENC_H
00071 #define JPC_T2ENC_H
00072
00073 /*****\
00074  * Includes.
00075  \*****/
00076
00077 #include "jasper/jas_stream.h"
00078
00079 #include "jpc_enc.h"
00080 #include "jpc_t2cod.h"
00081
00082 /*****\
00083  * Functions.
00084  \*****/
00085
00086 /* Encode the packets for a tile. */
00087 int jpc_enc_encpkts(jpc_enc_t *enc, jas_stream_t *out);
00088
00089 /* Encode the specified packet. */
00090 int jpc_enc_encpkt(jpc_enc_t *enc, jas_stream_t *out, unsigned compno,
00091 unsigned lvlno, unsigned prcno, unsigned lyrno);
00092
00093 /* Save the tier-2 coding state. */
00094 void jpc_save_t2state(jpc_enc_t *enc);
00095
00096 /* Restore the tier-2 coding state. */
00097 void jpc_restore_t2state(jpc_enc_t *enc);
00098
00099 /* Initialize the tier-2 coding state. */
00100 void jpc_init_t2state(jpc_enc_t *enc, bool raflag);
00101
00102 /* Create a packet iterator for the encoder. */
00103 jpc_pi_t *jpc_enc_pi_create(jpc_enc_cp_t *cp, jpc_enc_tile_t *tile);

```

```
00104
00105 #endif
```

16.66 jpc_tagtree.h

```
00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2002 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00064 /*
00065  * Tag Tree Library
00066  *
00067  * $Id$
00068  */
00069
00070 #ifndef JPC_TAGTREE_H
00071 #define JPC_TAGTREE_H
```

```

00072
00073 /*****
00074  * Includes
00075  *****/
00076
00077 #include <stdio.h>
00078
00079 #include "jpc_bs.h"
00080
00081 /*****
00082  * Constants
00083  *****/
00084
00085 /* The maximum allowable depth for a tag tree. */
00086 #define JPC_TAGTREE_MAXDEPTH    32
00087
00088 /*****
00089  * Types
00090  *****/
00091
00092 /*
00093  * Tag tree node.
00094  */
00095
00096 typedef struct jpc_tagtreenode_ {
00097
00098     /* The parent of this node. */
00099     struct jpc_tagtreenode_ *parent_;
00100
00101     /* The value associated with this node. */
00102     int value_;
00103
00104     /* The lower bound on the value associated with this node. */
00105     int low_;
00106
00107     /* A flag indicating if the value is known exactly. */
00108     int known_;
00109 } jpc_tagtreenode_t;
00110
00111 /*
00112  * Tag tree.
00113  */
00114
00115 typedef struct {
00116
00117     /* The number of leaves in the horizontal direction. */
00118     int numleafsh_;
00119
00120     /* The number of leaves in the vertical direction. */
00121     int numleafsv_;
00122
00123     /* The total number of nodes in the tree. */
00124     int numnodes_;
00125
00126     /* The nodes. */
00127     jpc_tagtreenode_t *nodes_;
00128 } jpc_tagtree_t;
00129
00130
00131 /*****
00132  * Functions.
00133  *****/
00134
00135 /* Create a tag tree. */
00136 jpc_tagtree_t *jpc_tagtree_create(int numleafsh, int numleafsv);
00137
00138 /* Destroy a tag tree. */
00139 void jpc_tagtree_destroy(jpc_tagtree_t *tree);
00140
00141 /* Copy data from one tag tree to another. */
00142 void jpc_tagtree_copy(jpc_tagtree_t *dsttree, const jpc_tagtree_t *srctree);
00143
00144 /* Reset the tag tree state. */
00145 void jpc_tagtree_reset(jpc_tagtree_t *tree);
00146
00147 /* Set the value associated with a particular leaf node of a tag tree. */
00148 void jpc_tagtree_setvalue(jpc_tagtree_t *tree, jpc_tagtreenode_t *leaf,
00149     int value);
00150
00151 /* Get a pointer to a particular leaf node. */

```

```

00153 JAS_ATTRIBUTE_PURE
00154 jpc_tagreenode_t *jpc_tagtree_getleaf(jpc_tagtree_t *tree, int n);
00155
00156 /* Invoke the tag tree decoding procedure. */
00157 int jpc_tagtree_decode(jpc_tagtree_t *tree, jpc_tagreenode_t *leaf,
00158     int threshold, jpc_bitstream_t *in);
00159
00160 /* Invoke the tag tree encoding procedure. */
00161 int jpc_tagtree_encode(jpc_tagtree_t *tree, jpc_tagreenode_t *leaf,
00162     int threshold, jpc_bitstream_t *out);
00163
00164 /* Dump a tag tree (for debugging purposes). */
00165 void jpc_tagtree_dump(const jpc_tagtree_t *tree, FILE *out);
00166
00167 #endif

```

16.67 jpc_tsfb.h

```

00001 /*
00002  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003  *   British Columbia.
00004  * Copyright (c) 2001-2004 Michael David Adams.
00005  * All rights reserved.
00006  */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009  *
00010  * JasPer License Version 2.0
00011  *
00012  * Copyright (c) 2001-2006 Michael David Adams
00013  * Copyright (c) 1999-2000 Image Power, Inc.
00014  * Copyright (c) 1999-2000 The University of British Columbia
00015  *
00016  * All rights reserved.
00017  *
00018  * Permission is hereby granted, free of charge, to any person (the
00019  * "User") obtaining a copy of this software and associated documentation
00020  * files (the "Software"), to deal in the Software without restriction,
00021  * including without limitation the rights to use, copy, modify, merge,
00022  * publish, distribute, and/or sell copies of the Software, and to permit
00023  * persons to whom the Software is furnished to do so, subject to the
00024  * following conditions:
00025  *
00026  * 1. The above copyright notices and this permission notice (which
00027  * includes the disclaimer below) shall be included in all copies or
00028  * substantial portions of the Software.
00029  *
00030  * 2. The name of a copyright holder shall not be used to endorse or
00031  * promote products derived from the Software without specific prior
00032  * written permission.
00033  *
00034  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00044  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY

```

```

00059  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060  *
00061  * __END_OF_JASPER_LICENSE__
00062  */
00063
00064 /*
00065  * Tree-Structured Filter Bank (TSFB) Library
00066  *
00067  * $Id$
00068  */
00069
00070 #ifndef JPC_TSFB_H
00071 #define JPC_TSFB_H
00072
00073 /*****\
00074  * Includes.
00075  \*****/
00076
00077 #include "jasper/jas_seq.h"
00078 #include "jasper/jas_types.h"
00079
00080 #include "jpc_fix.h"
00081 #include "jpc_qmfb.h"
00082
00083 /*****\
00084  * Constants.
00085  \*****/
00086
00087 #define JPC_TSFB_MAXBANDS      (JPC_TSFB_MAXDEPTH * 3 + 1)
00088 #define JPC_TSFB_MAXDEPTH     32
00089 #define JPC_TSFB_RITIMODE     JPC_QMFB1D_RITIMODE
00090
00091 enum jpc_tsfb_orient {
00092     JPC_TSFB_LL = 0,
00093     JPC_TSFB_LH = 1,
00094     JPC_TSFB_HL = 2,
00095     JPC_TSFB_HH = 3,
00096 };
00097
00098 /*****\
00099  * Types.
00100  \*****/
00101
00102 typedef struct {
00103     int xstart;
00104     int ystart;
00105     int xend;
00106     int yend;
00107     enum jpc_tsfb_orient orient;
00108     int locxstart;
00109     int locystart;
00110     int locxend;
00111     int locyend;
00112     jpc_fix_t synenergywt;
00113 } jpc_tsfb_band_t;
00114
00115 typedef struct {
00116     unsigned numlvl;
00117     const jpc_qmfb2d_t *qmfb;
00118 } jpc_tsfb_t;
00119
00120 /*****\
00121  * Functions.
00122  \*****/
00123
00124 /* Create a TSFB. */
00125 jpc_tsfb_t *jpc_cod_gettsfb(unsigned qmfbid, unsigned numlevels);
00126
00127 /* Destroy a TSFB. */
00128 void jpc_tsfb_destroy(jpc_tsfb_t *tsfb);
00129
00130 /* Perform analysis. */
00131 int jpc_tsfb_analyze(jpc_tsfb_t *tsfb, jas_seq2d_t *x);
00132
00133 /* Perform synthesis. */
00134 int jpc_tsfb_synthesize(jpc_tsfb_t *tsfb, jas_seq2d_t *x);
00135
00136 /* Get band information for a TSFB. */
00137 int jpc_tsfb_getbands(jpc_tsfb_t *tsfb, uint_fast32_t xstart,
00138     uint_fast32_t ystart, uint_fast32_t xend, uint_fast32_t yend,
00139     jpc_tsfb_band_t *bands);

```

```
00140
00141 #endif
```

16.68 jpc_util.h

```
00001 /*
00002  * Copyright (c) 2001-2002 Michael David Adams.
00003  * All rights reserved.
00004  */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007  *
00008  * JasPer License Version 2.0
00009  *
00010  * Copyright (c) 2001-2006 Michael David Adams
00011  * Copyright (c) 1999-2000 Image Power, Inc.
00012  * Copyright (c) 1999-2000 The University of British Columbia
00013  *
00014  * All rights reserved.
00015  *
00016  * Permission is hereby granted, free of charge, to any person (the
00017  * "User") obtaining a copy of this software and associated documentation
00018  * files (the "Software"), to deal in the Software without restriction,
00019  * including without limitation the rights to use, copy, modify, merge,
00020  * publish, distribute, and/or sell copies of the Software, and to permit
00021  * persons to whom the Software is furnished to do so, subject to the
00022  * following conditions:
00023  *
00024  * 1. The above copyright notices and this permission notice (which
00025  * includes the disclaimer below) shall be included in all copies or
00026  * substantial portions of the Software.
00027  *
00028  * 2. The name of a copyright holder shall not be used to endorse or
00029  * promote products derived from the Software without specific prior
00030  * written permission.
00031  *
00032  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00054  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058  *
00059  * __END_OF_JASPER_LICENSE__
00060  */
00061
00062 #ifndef JPC_UTIL_H
00063 #define JPC_UTIL_H
00064
00065 #include "jpc_fix.h"
00066
00067 #include "jasper/jas_seq.h"
00068
00069 /* Parse a comma separated list of real numbers into an array of doubles. */
00070 int jpc_atoaf(const char *s, int *numvalues, double **values);
00071
```

```

00072 /* Upsample a sequence. */
00073 jas_seq_t *jpc_seq_upsample(const jas_seq_t *seq, int n);
00074
00075 /* Convolve two sequences. */
00076 jas_seq_t *jpc_seq_conv(const jas_seq_t *seq0, const jas_seq_t *seq1);
00077
00078 /* Compute the norm of a sequence. */
00079 JAS_ATTRIBUTE_PURE
00080 jpc_fix_t jpc_seq_norm(const jas_seq_t *x);
00081
00082 #endif

```

16.69 jpg_cod.h

```

00001 /*
00002  * Copyright (c) 2001-2002 Michael David Adams.
00003  * All rights reserved.
00004  */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007  *
00008  * JasPer License Version 2.0
00009  *
00010  * Copyright (c) 2001-2006 Michael David Adams
00011  * Copyright (c) 1999-2000 Image Power, Inc.
00012  * Copyright (c) 1999-2000 The University of British Columbia
00013  *
00014  * All rights reserved.
00015  *
00016  * Permission is hereby granted, free of charge, to any person (the
00017  * "User") obtaining a copy of this software and associated documentation
00018  * files (the "Software"), to deal in the Software without restriction,
00019  * including without limitation the rights to use, copy, modify, merge,
00020  * publish, distribute, and/or sell copies of the Software, and to permit
00021  * persons to whom the Software is furnished to do so, subject to the
00022  * following conditions:
00023  *
00024  * 1. The above copyright notices and this permission notice (which
00025  * includes the disclaimer below) shall be included in all copies or
00026  * substantial portions of the Software.
00027  *
00028  * 2. The name of a copyright holder shall not be used to endorse or
00029  * promote products derived from the Software without specific prior
00030  * written permission.
00031  *
00032  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00054  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058  *
00059  * __END_OF_JASPER_LICENSE__
00060  */
00061
00062 /*

```

```

00063  * JPG Format Library
00064  *
00065  * $Id$
00066  */
00067
00068 #ifndef JPG_COD_H
00069 #define JPG_COD_H
00070
00071 /*****\
00072 * Includes.
00073 \*****/
00074
00075 /*****\
00076 * Constants.
00077 \*****/
00078
00079 #define JPG_MAGIC      0xffd8
00080 #define JPG_MAGICLEN 2
00081
00082 #endif

```

16.70 jpg_enc.h

```

00001 /*
00002  * Copyright (c) 2002 Michael David Adams.
00003  * All rights reserved.
00004  */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007  *
00008  * JasPer License Version 2.0
00009  *
00010  * Copyright (c) 2001-2006 Michael David Adams
00011  * Copyright (c) 1999-2000 Image Power, Inc.
00012  * Copyright (c) 1999-2000 The University of British Columbia
00013  *
00014  * All rights reserved.
00015  *
00016  * Permission is hereby granted, free of charge, to any person (the
00017  * "User") obtaining a copy of this software and associated documentation
00018  * files (the "Software"), to deal in the Software without restriction,
00019  * including without limitation the rights to use, copy, modify, merge,
00020  * publish, distribute, and/or sell copies of the Software, and to permit
00021  * persons to whom the Software is furnished to do so, subject to the
00022  * following conditions:
00023  *
00024  * 1. The above copyright notices and this permission notice (which
00025  * includes the disclaimer below) shall be included in all copies or
00026  * substantial portions of the Software.
00027  *
00028  * 2. The name of a copyright holder shall not be used to endorse or
00029  * promote products derived from the Software without specific prior
00030  * written permission.
00031  *
00032  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH

```



```

00054 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058 *
00059 * __END_OF_JASPER_LICENSE__
00060 */
00061
00062 #ifndef JPG_ENC_H
00063 #define JPG_ENC_H
00064
00065 typedef struct {
00066     int numcmpts;
00067     int cmpts[4];
00068 } jpg_enc_t;
00069
00070 #endif

```

16.71 jpg_jpeglib.h

```

00001 /*
00002 * Copyright (c) 2001-2002 Michael David Adams.
00003 * All rights reserved.
00004 */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007 *
00008 * JasPer License Version 2.0
00009 *
00010 * Copyright (c) 2001-2006 Michael David Adams
00011 * Copyright (c) 1999-2000 Image Power, Inc.
00012 * Copyright (c) 1999-2000 The University of British Columbia
00013 *
00014 * All rights reserved.
00015 *
00016 * Permission is hereby granted, free of charge, to any person (the
00017 * "User") obtaining a copy of this software and associated documentation
00018 * files (the "Software"), to deal in the Software without restriction,
00019 * including without limitation the rights to use, copy, modify, merge,
00020 * publish, distribute, and/or sell copies of the Software, and to permit
00021 * persons to whom the Software is furnished to do so, subject to the
00022 * following conditions:
00023 *
00024 * 1. The above copyright notices and this permission notice (which
00025 * includes the disclaimer below) shall be included in all copies or
00026 * substantial portions of the Software.
00027 *
00028 * 2. The name of a copyright holder shall not be used to endorse or
00029 * promote products derived from the Software without specific prior
00030 * written permission.
00031 *
00032 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00054 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY

```

```

00057  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058  *
00059  * __END_OF_JASPER_LICENSE__
00060  */
00061
00062 #ifndef JPG_JPEGLIB_H
00063 #define JPG_JPEGLIB_H
00064
00065 /*****\
00066 * Includes.
00067 \*****/
00068
00069 #include <stdio.h>
00070 #include "jasper/jas_types.h"
00071
00072 /* Note: The jpeglib.h header file does not include definitions of
00073    FILE, size_t, etc. */
00074 #include <jpeglib.h> /* IWYU pragma: export */
00075
00076 #endif

```

16.72 mif_cod.h

```

00001 /*
00002  * Copyright (c) 2001-2002 Michael David Adams.
00003  * All rights reserved.
00004  */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007  *
00008  * JasPer License Version 2.0
00009  *
00010  * Copyright (c) 2001-2006 Michael David Adams
00011  * Copyright (c) 1999-2000 Image Power, Inc.
00012  * Copyright (c) 1999-2000 The University of British Columbia
00013  *
00014  * All rights reserved.
00015  *
00016  * Permission is hereby granted, free of charge, to any person (the
00017  * "User") obtaining a copy of this software and associated documentation
00018  * files (the "Software"), to deal in the Software without restriction,
00019  * including without limitation the rights to use, copy, modify, merge,
00020  * publish, distribute, and/or sell copies of the Software, and to permit
00021  * persons to whom the Software is furnished to do so, subject to the
00022  * following conditions:
00023  *
00024  * 1. The above copyright notices and this permission notice (which
00025  * includes the disclaimer below) shall be included in all copies or
00026  * substantial portions of the Software.
00027  *
00028  * 2. The name of a copyright holder shall not be used to endorse or
00029  * promote products derived from the Software without specific prior
00030  * written permission.
00031  *
00032  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH

```

```

00054 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058 *
00059 * __END_OF_JASPER_LICENSE__
00060 */
00061
00062 #ifndef MIF_COD_H
00063 #define MIF_COD_H
00064
00065 /*****\
00066 * Includes.
00067 \*****/
00068
00069 #include "jasper/jas_types.h"
00070
00071 /*****\
00072 * Constants.
00073 \*****/
00074
00075 #define MIF_MAGIC                0x4d49460a
00076 /* signature */
00077
00078 #define MIF_MAGICLEN            4
00079 /* length of signature in bytes */
00080
00081 /*****\
00082 * Types.
00083 \*****/
00084
00085 /* Per-component information. */
00086
00087 typedef struct {
00088
00089     int_fast32_t tlx;
00090
00091     int_fast32_t tly;
00092
00093     int_fast32_t width;
00094
00095     int_fast32_t height;
00096
00097     int_fast32_t sampperx;
00098
00099     int_fast32_t samppery;
00100
00101     int_fast16_t prec;
00102
00103     int_fast16_t sgnd;
00104
00105     char *data;
00106
00107 } mif_cmpt_t;
00108
00109 /* MIF header. */
00110
00111 typedef struct {
00112
00113     uint_fast32_t magic;
00114
00115     int numcmpts;
00116
00117     int maxcmpts;
00118
00119     mif_cmpt_t **cmpts;
00120
00121 } mif_hdr_t;
00122
00123 #endif

```

16.73 pgx_cod.h

```

00001 /*
00002 * Copyright (c) 2001-2002 Michael David Adams.
00003 * All rights reserved.

```

```

00004 */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007 *
00008 * JasPer License Version 2.0
00009 *
00010 * Copyright (c) 2001-2006 Michael David Adams
00011 * Copyright (c) 1999-2000 Image Power, Inc.
00012 * Copyright (c) 1999-2000 The University of British Columbia
00013 *
00014 * All rights reserved.
00015 *
00016 * Permission is hereby granted, free of charge, to any person (the
00017 * "User") obtaining a copy of this software and associated documentation
00018 * files (the "Software"), to deal in the Software without restriction,
00019 * including without limitation the rights to use, copy, modify, merge,
00020 * publish, distribute, and/or sell copies of the Software, and to permit
00021 * persons to whom the Software is furnished to do so, subject to the
00022 * following conditions:
00023 *
00024 * 1. The above copyright notices and this permission notice (which
00025 * includes the disclaimer below) shall be included in all copies or
00026 * substantial portions of the Software.
00027 *
00028 * 2. The name of a copyright holder shall not be used to endorse or
00029 * promote products derived from the Software without specific prior
00030 * written permission.
00031 *
00032 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00054 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058 *
00059 * __END_OF_JASPER_LICENSE__
00060 */
00061
00062 /*
00063 * PGX Format Library
00064 *
00065 * $Id$
00066 */
00067
00068 #ifndef PGX_COD_H
00069 #define PGX_COD_H
00070
00071 /*****\
00072 * Includes.
00073 \*****/
00074
00075 #include <stdio.h>
00076
00077 #include "jasper/jas_types.h"
00078
00079 /*****\
00080 * Constants.
00081 \*****/
00082
00083 #define PGX_MAGIC 0x5047
00084 #define PGX_MAGICLEN 2

```

```

00085
00086 /*****
00087  * Types.
00088  *****/
00089
00090 typedef struct {
00091     uint_fast16_t magic;
00092     /* The signature. */
00093
00094     bool bigendian;
00095     /* The byte ordering used. */
00096
00097     bool sgnd;
00098     /* The signedness of the samples. */
00099
00100     uint_fast32_t prec;
00101     /* The precision of the samples. */
00102
00103     uint_fast32_t width;
00104     /* The width of the component. */
00105
00106     uint_fast32_t height;
00107     /* The height of the component. */
00108
00109 } pgx_hdr_t;
00110
00111 /*****
00112  * Functions.
00113  *****/
00114
00115 void pgx_dumphdr(FILE *out, pgx_hdr_t *hdr);
00116
00117 #endif

```

16.74 pgx_enc.h

```

00001 /*
00002  * Copyright (c) 2002 Michael David Adams.
00003  * All rights reserved.
00004  */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007  *
00008  * JasPer License Version 2.0
00009  *
00010  * Copyright (c) 2001-2006 Michael David Adams
00011  * Copyright (c) 1999-2000 Image Power, Inc.
00012  * Copyright (c) 1999-2000 The University of British Columbia
00013  *
00014  * All rights reserved.
00015  *
00016  * Permission is hereby granted, free of charge, to any person (the
00017  * "User") obtaining a copy of this software and associated documentation
00018  * files (the "Software"), to deal in the Software without restriction,
00019  * including without limitation the rights to use, copy, modify, merge,
00020  * publish, distribute, and/or sell copies of the Software, and to permit
00021  * persons to whom the Software is furnished to do so, subject to the
00022  * following conditions:
00023  *
00024  * 1. The above copyright notices and this permission notice (which
00025  * includes the disclaimer below) shall be included in all copies or
00026  * substantial portions of the Software.
00027  *
00028  * 2. The name of a copyright holder shall not be used to endorse or
00029  * promote products derived from the Software without specific prior
00030  * written permission.
00031  *
00032  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING

```

```

00040 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00054 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058 *
00059 * __END_OF_JASPER_LICENSE__
00060 */
00061
00062 #ifndef PGX_ENC_H
00063 #define PGX_ENC_H
00064
00065 typedef struct {
00066     int cmpt;
00067 } pgx_enc_t;
00068
00069 #endif

```

16.75 pnm_cod.h

```

00001 /*
00002 * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003 * British Columbia.
00004 * Copyright (c) 2001-2002 Michael David Adams.
00005 * All rights reserved.
00006 */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009 *
00010 * JasPer License Version 2.0
00011 *
00012 * Copyright (c) 2001-2006 Michael David Adams
00013 * Copyright (c) 1999-2000 Image Power, Inc.
00014 * Copyright (c) 1999-2000 The University of British Columbia
00015 *
00016 * All rights reserved.
00017 *
00018 * Permission is hereby granted, free of charge, to any person (the
00019 * "User") obtaining a copy of this software and associated documentation
00020 * files (the "Software"), to deal in the Software without restriction,
00021 * including without limitation the rights to use, copy, modify, merge,
00022 * publish, distribute, and/or sell copies of the Software, and to permit
00023 * persons to whom the Software is furnished to do so, subject to the
00024 * following conditions:
00025 *
00026 * 1. The above copyright notices and this permission notice (which
00027 * includes the disclaimer below) shall be included in all copies or
00028 * substantial portions of the Software.
00029 *
00030 * 2. The name of a copyright holder shall not be used to endorse or
00031 * promote products derived from the Software without specific prior
00032 * written permission.
00033 *
00034 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION

```

```

00044 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060 *
00061 * __END_OF_JASPER_LICENSE__
00062 */
00063
00064 /*
00065  * Portable Pixmap/Graymap Format Support
00066  *
00067  * $Id$
00068  */
00069
00070 #ifndef PNM_COD_H
00071 #define PNM_COD_H
00072
00073 /*****\
00074  * Includes.
00075  \*****/
00076
00077 #include "jasper/jas_types.h"
00078
00079 /*****\
00080  * Constants.
00081  \*****/
00082
00083 /* Magic numbers. */
00084 #define PNM_MAGIC_TXTPBM 0x5031 /* Text Portable BitMap (P1) */
00085 #define PNM_MAGIC_TXTPGM 0x5032 /* Text Portable GrayMap (P2) */
00086 #define PNM_MAGIC_TXTPPM 0x5033 /* Text Portable PixMap (P3) */
00087 #define PNM_MAGIC_BINPBM 0x5034 /* Binary Portable BitMap (P4) */
00088 #define PNM_MAGIC_BINPGM 0x5035 /* Binary Portable GrayMap (P5) */
00089 #define PNM_MAGIC_BINPPM 0x5036 /* Binary Portable PixMap (P6) */
00090 #define PNM_MAGIC_PAM 0x5037 /* PAM (P7) */
00091
00092 /* Type of image data. */
00093 #define PNM_TYPE_INVALID 0
00094 #define PNM_TYPE_PBM 1 /* BitMap */
00095 #define PNM_TYPE_PGM 2 /* GrayMap */
00096 #define PNM_TYPE_PPM 3 /* PixMap */
00097
00098 /* Format of image data. */
00099 #define PNM_FMT_TXT 0 /* Text */
00100 #define PNM_FMT_BIN 1 /* Binary */
00101
00102 #define PNM_MAXLINELEN 79
00103
00104 #define PNM_TUPLETYPE_UNKNOWN 0
00105 #define PNM_TUPLETYPE_MONO 1
00106 #define PNM_TUPLETYPE_GRAY 2
00107 #define PNM_TUPLETYPE_GRAYA 3
00108 #define PNM_TUPLETYPE_RGB 4
00109 #define PNM_TUPLETYPE_RGBA 5
00110
00111 /*****\
00112  * Types.
00113  \*****/
00114
00115 /* File header. */
00116
00117 typedef struct {
00118
00119     int magic;
00120     /* The magic number. */
00121
00122     int width;
00123     /* The image width. */
00124

```

```

00125         int height;
00126         /* The image height. */
00127
00128         int numcmpts;
00129
00130         int maxval;
00131         /* The maximum allowable sample value. */
00132
00133 #if 0
00134         int tupletype;
00135 #endif
00136
00137         bool sgnd;
00138         /* The sample data is signed. */
00139
00140 } pnm_hdr_t;
00141
00142 /*****
00143  * Functions.
00144  *****/
00145
00146 int pnm_type(uint_fast16_t magic);
00147 /* Determine type (i.e., PGM or PPM) from magic number. */
00148
00149 int pnm_fmt(uint_fast16_t magic);
00150 /* Determine format (i.e., text or binary) from magic number. */
00151
00152 int pnm_maxvaltodepth(uint_fast32_t maxval);
00153 /* Determine depth (i.e., precision) from maximum value. */
00154
00155 #define PNM_ONES(n) \
00156     ((n) < 32) ? ((1UL << (n)) - 1) : 0xffffffffUL
00157 #endif

```

16.76 pnm_enc.h

```

00001 /*
00002  * Copyright (c) 2002 Michael David Adams.
00003  * All rights reserved.
00004  */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007  *
00008  * JasPer License Version 2.0
00009  *
00010  * Copyright (c) 2001-2006 Michael David Adams
00011  * Copyright (c) 1999-2000 Image Power, Inc.
00012  * Copyright (c) 1999-2000 The University of British Columbia
00013  *
00014  * All rights reserved.
00015  *
00016  * Permission is hereby granted, free of charge, to any person (the
00017  * "User") obtaining a copy of this software and associated documentation
00018  * files (the "Software"), to deal in the Software without restriction,
00019  * including without limitation the rights to use, copy, modify, merge,
00020  * publish, distribute, and/or sell copies of the Software, and to permit
00021  * persons to whom the Software is furnished to do so, subject to the
00022  * following conditions:
00023  *
00024  * 1. The above copyright notices and this permission notice (which
00025  * includes the disclaimer below) shall be included in all copies or
00026  * substantial portions of the Software.
00027  *
00028  * 2. The name of a copyright holder shall not be used to endorse or
00029  * promote products derived from the Software without specific prior
00030  * written permission.
00031  *
00032  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00034  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,

```



```

00041 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00054 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058 *
00059 * __END_OF_JASPER_LICENSE__
00060 */
00061
00062 #ifndef PNM_ENC_H
00063 #define PNM_ENC_H
00064
00065 typedef struct {
00066     int numcmpts;
00067     int cmpts[4];
00068 } pnm_enc_t;
00069
00070 #endif

```

16.77 ras_cod.h

```

00001 /*
00002 * Copyright (c) 1999-2000 Image Power, Inc. and the University of
00003 * British Columbia.
00004 * Copyright (c) 2001-2002 Michael David Adams.
00005 * All rights reserved.
00006 */
00007
00008 /* __START_OF_JASPER_LICENSE__
00009 *
00010 * JasPer License Version 2.0
00011 *
00012 * Copyright (c) 2001-2006 Michael David Adams
00013 * Copyright (c) 1999-2000 Image Power, Inc.
00014 * Copyright (c) 1999-2000 The University of British Columbia
00015 *
00016 * All rights reserved.
00017 *
00018 * Permission is hereby granted, free of charge, to any person (the
00019 * "User") obtaining a copy of this software and associated documentation
00020 * files (the "Software"), to deal in the Software without restriction,
00021 * including without limitation the rights to use, copy, modify, merge,
00022 * publish, distribute, and/or sell copies of the Software, and to permit
00023 * persons to whom the Software is furnished to do so, subject to the
00024 * following conditions:
00025 *
00026 * 1. The above copyright notices and this permission notice (which
00027 * includes the disclaimer below) shall be included in all copies or
00028 * substantial portions of the Software.
00029 *
00030 * 2. The name of a copyright holder shall not be used to endorse or
00031 * promote products derived from the Software without specific prior
00032 * written permission.
00033 *
00034 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00035 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
00036 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00037 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00038 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00039 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00040 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00041 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00042 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00043 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION

```

```

00044 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00045 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00046 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00047 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00048 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00049 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00050 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00051 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00052 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00053 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00054 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00055 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00056 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00057 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00058 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00059 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00060 *
00061 * __END_OF_JASPER_LICENSE__
00062 */
00063
00064 /*
00065  * Sun Rasterfile Library
00066  *
00067  * $Id$
00068  */
00069
00070 /*****
00071  * Sun Rasterfile
00072  \*****/
00073
00074 #ifndef RAS_COD_H
00075 #define RAS_COD_H
00076
00077 /*****
00078  * Includes.
00079  \*****/
00080
00081 #include "jasper/jas_types.h"
00082
00083 /*****
00084  * Constants.
00085  \*****/
00086
00087 #define RAS_MAGIC      0x59a66a95
00088 #define RAS_MAGICLEN   4
00089
00090 #define RAS_TYPE_OLD   0
00091 #define RAS_TYPE_STD   1
00092 #define RAS_TYPE_RLE   2
00093
00094 #define RAS_MT_NONE    0
00095 #define RAS_MT_EQUALRGB 1
00096 #define RAS_MT_RAW     2
00097
00098 /*****
00099  * Types.
00100  \*****/
00101
00102 /* Sun Rasterfile header. */
00103 typedef struct {
00104
00105     int_fast32_t magic;
00106     /* signature */
00107
00108     int_fast32_t width;
00109     /* width of image (in pixels) */
00110
00111     int_fast32_t height;
00112     /* height of image (in pixels) */
00113
00114     int_fast32_t depth;
00115     /* number of bits per pixel */
00116
00117     int_fast32_t length;
00118     /* length of image data (in bytes) */
00119
00120     int_fast32_t type;
00121     /* format of image data */
00122
00123     int_fast32_t maptype;
00124     /* colormap type */

```

```

00125
00126     int_fast32_t maplength;
00127     /* length of colormap data (in bytes) */
00128 } ras_hdr_t;
00129
00130
00131 #define RAS_CMAP_MAXSIZE 256
00132
00133 /* Color map. */
00134 typedef struct {
00135     int len;
00136     /* The number of entries in the color map. */
00137
00138     int data[RAS_CMAP_MAXSIZE];
00139     /* The color map data. */
00140 } ras_cmap_t;
00141
00142
00143
00144 /*****
00145  * Macros.
00146  *****/
00147
00148 #define RAS_GETBLUE(x)  (((x) >> 16) & 0xff)
00149 #define RAS_GETGREEN(x) (((x) >> 8) & 0xff)
00150 #define RAS_GETRED(x)   ((x) & 0xff)
00151
00152 #define RAS_BLUE(x)      (((x) & 0xff) << 16)
00153 #define RAS_GREEN(x)     (((x) & 0xff) << 8)
00154 #define RAS_RED(x)       ((x) & 0xff)
00155
00156 #define RAS_ROWSIZE(hdr) \
00157     (((hdr)->width * (hdr)->depth + 15) / 16) * 2)
00158 #define RAS_ISRGB(hdr) \
00159     ((hdr)->depth == 24 || (hdr)->depth == 32)
00160
00161 #define RAS_ONES(n) \
00162     (((n) == 32) ? 0xffffffffUL : ((1UL << (n)) - 1))
00163
00164 #endif

```

16.78 ras_enc.h

```

00001 /*
00002  * Copyright (c) 2002 Michael David Adams.
00003  * All rights reserved.
00004  */
00005
00006 /* __START_OF_JASPER_LICENSE__
00007  *
00008  * JasPer License Version 2.0
00009  *
00010  * Copyright (c) 2001-2006 Michael David Adams
00011  * Copyright (c) 1999-2000 Image Power, Inc.
00012  * Copyright (c) 1999-2000 The University of British Columbia
00013  *
00014  * All rights reserved.
00015  *
00016  * Permission is hereby granted, free of charge, to any person (the
00017  * "User") obtaining a copy of this software and associated documentation
00018  * files (the "Software"), to deal in the Software without restriction,
00019  * including without limitation the rights to use, copy, modify, merge,
00020  * publish, distribute, and/or sell copies of the Software, and to permit
00021  * persons to whom the Software is furnished to do so, subject to the
00022  * following conditions:
00023  *
00024  * 1. The above copyright notices and this permission notice (which
00025  * includes the disclaimer below) shall be included in all copies or
00026  * substantial portions of the Software.
00027  *
00028  * 2. The name of a copyright holder shall not be used to endorse or
00029  * promote products derived from the Software without specific prior
00030  * written permission.
00031  *
00032  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
00033  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER

```

```
00034 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
00035 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
00036 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
00037 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
00038 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
00039 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
00040 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
00041 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
00042 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
00043 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
00044 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
00045 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
00046 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
00047 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
00048 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
00049 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
00050 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
00051 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
00052 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
00053 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
00054 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
00055 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
00056 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
00057 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
00058 *
00059 * __END_OF_JASPER_LICENSE__
00060 */
00061
00062 #ifndef RAS_ENC_H
00063 #define RAS_ENC_H
00064
00065 typedef struct {
00066     int numcmpts;
00067     int cmpts[4];
00068 } ras_enc_t;
00069
00070 #endif
```