

JasPer

Generated by Doxygen 1.9.3

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)
- 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)
- non-native support for following image codecs:
 - JPEG (ISO/IEC 10918-1) via IJG JPEG Library
 - HEIC via the libheif Library
- can add new codecs (or enable/disable codecs) at run time
- can specify memory allocator to be used by library
- can specify logging function to be used by library for error, warning, informational, and debugging messages

- can place upper limit on total amount of memory used by 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 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>

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)
* 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 to 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_DTYPETOBOPC`, `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.

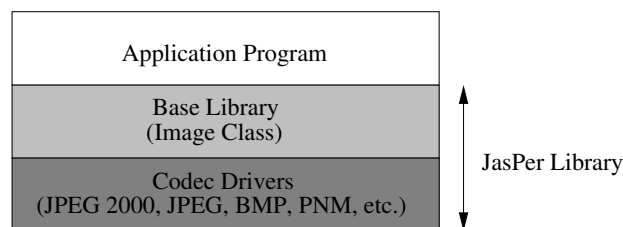


Figure 5.1 Software structure

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.

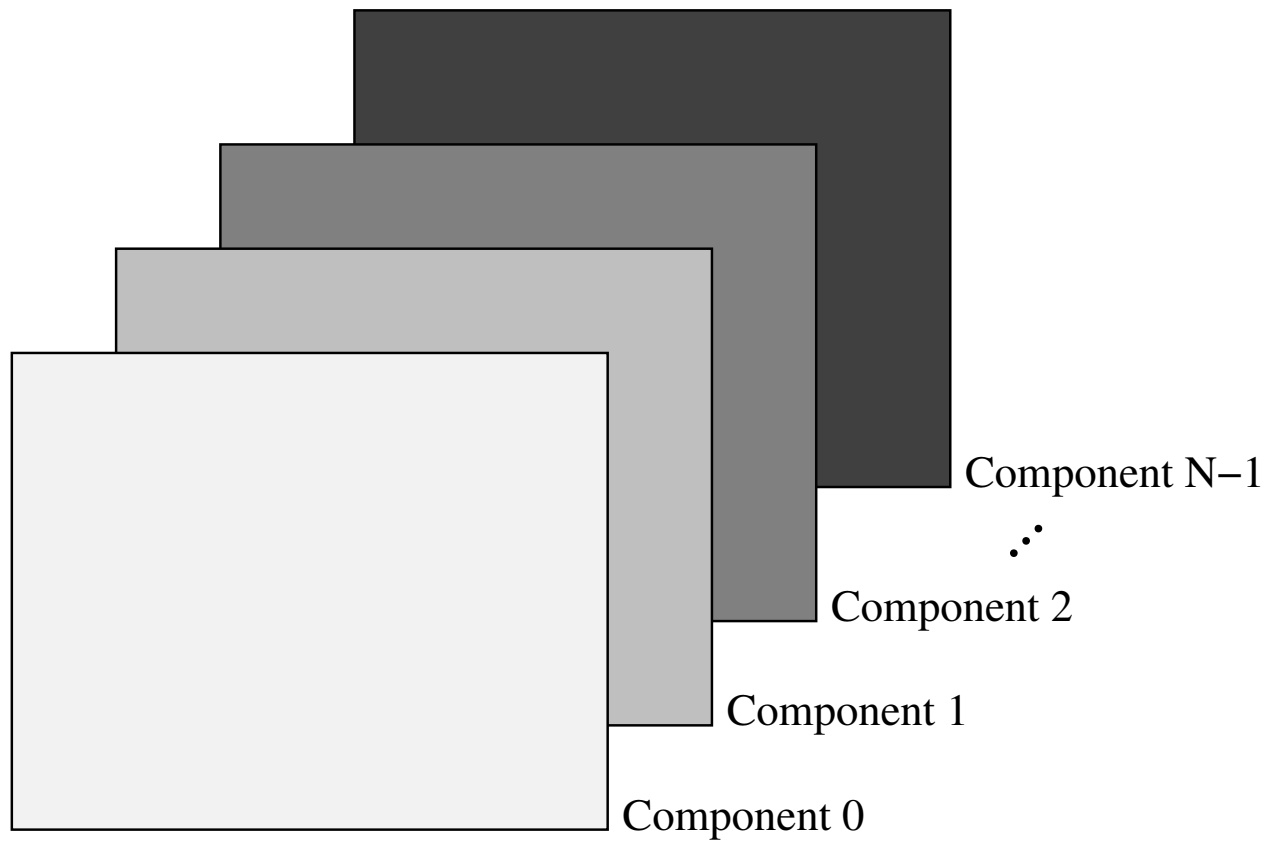


Figure 5.2 Image model: An image with N components.

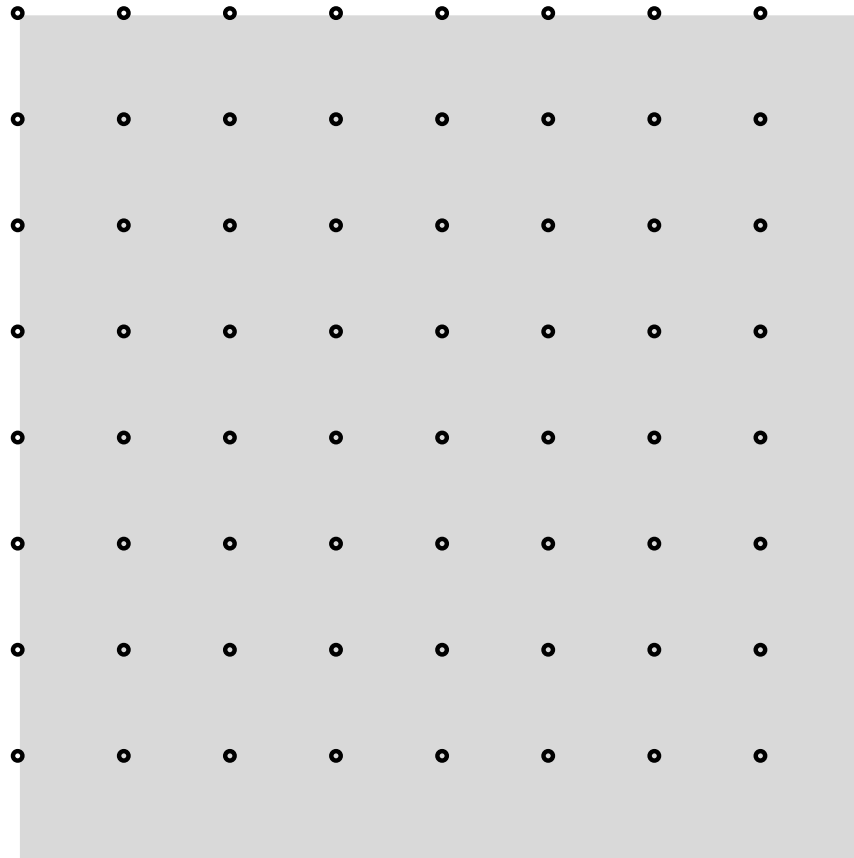


Figure 5.3 Image model: An individual component.

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.

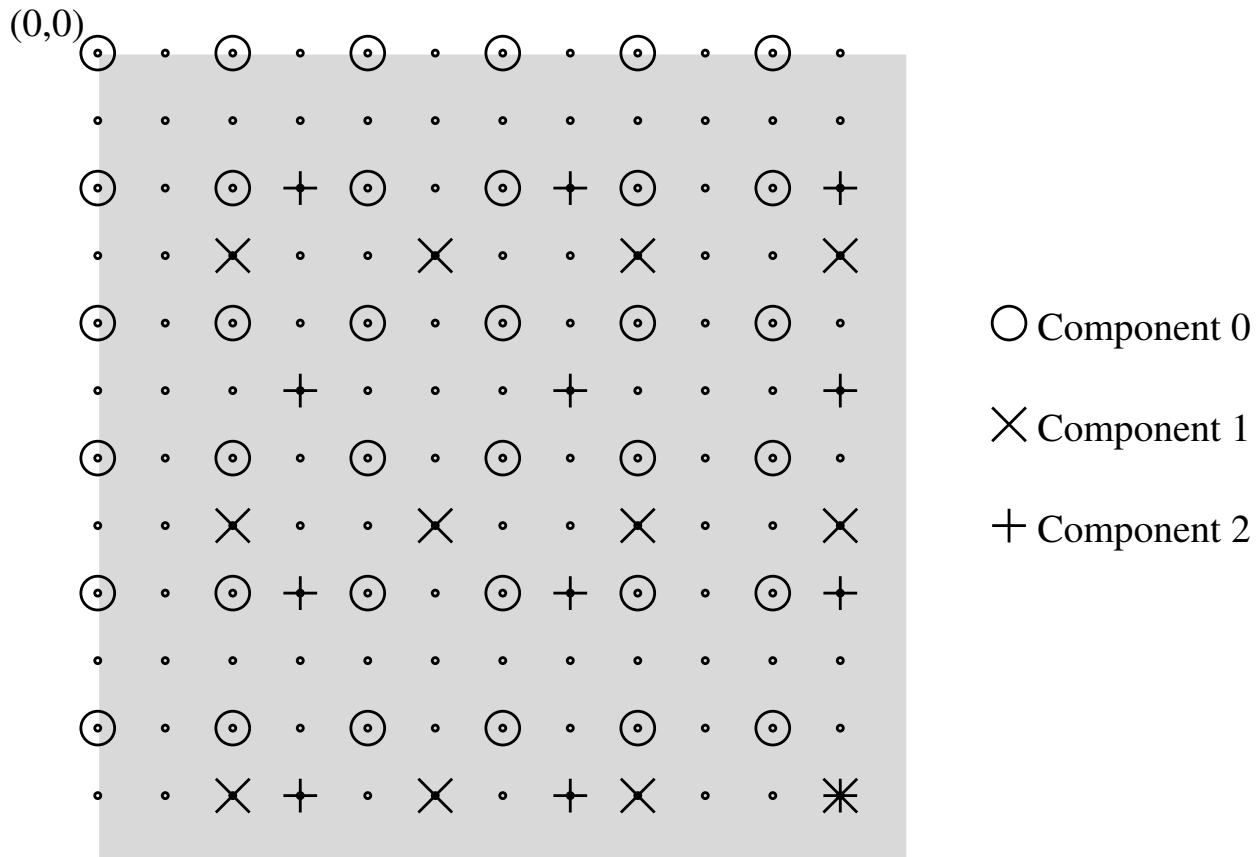


Figure 5.4 Alignment of components on the reference grid.

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 cleanup 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.
- `--wait $n`
 - Automatically step from one image to the next, pausing for `$n` seconds in between.

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:

- `-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

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

Module Index

11.1 Modules

Here is a list of all modules:

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_cmcmptfmt_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

Module 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()`

```
static 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()`

```
static 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()`

```
static 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

<code>max_mem</code>	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 )
```

Initialize a memory allocator that uses malloc and related functions for managing memory.

Parameters

<code>allocator</code>	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)` (((stream)->flags_ & JAS_STREAM_EOF) != 0)
Get the EOF indicator for a stream.
- #define `jas_stream_error(stream)` (((stream)->flags_ & JAS_STREAM_ERR) != 0)
Get the error indicator for a stream.
- #define `jas_stream_clearerr(stream)` ((stream)->flags_ &= ~(JAS_STREAM_ERR | JAS_STREAM_EOF))
Clear the error indicator for a stream.
- #define `jas_stream_getrwlmit(stream)` (((const `jas_stream_t` *) (stream))->rwlmit_)
Get the read/write limit for a stream.

- `#define jas_stream_getrwcoun(stream) (((const jas_stream_t*)(stream))->rwcnt_)`
Get the read/write count for a stream.
- `#define jas_stream_getc(stream) jas_stream_getc_func(stream)`
jas_stream_getc Read a character from a stream.
- `#define jas_stream_putc(stream, c) jas_stream_putc_func(stream, c)`
jas_stream_putc Write a character to a stream.
- `#define jas_stream_peekc(stream)`
Look at the next character to be read from a stream without actually removing the character from the stream.

Functions

- `JAS_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_coun)`
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 )    ((stream)->flags_ &= ~(JAS_STREAM_ERR | JAS_STREAM_EOF))
```

Clear the error indicator for a stream.

Parameters

stream	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 )  (((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 )  (((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 )  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_getrwcoun

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

Get the read/write count for a stream.

Parameters

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

Returns

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

14.3.2.6 jas_stream_getrwlmit

```
#define jas_stream_getrwlmit(  
    stream )    (((const jas_stream_t *) (stream))->rwlmit_)
```

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 ) 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 )
```

Open a file descriptor as a stream.

Parameters

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

Returns

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

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>fdopen</code> function in the C standard library.
<i>fp</i>	A pointer to the <code>FILE</code> (i.e., <code>stdio</code> 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 (
    jas_stream_t * stream,
    size_t count,
    int value )
```

Write a fill character multiple times to a stream.

Parameters

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

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

Returns

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

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.

14.3.3.20 `jas_stream_seek()`

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

Set the current position within the stream.

Parameters

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

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

Returns

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

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>rwlmit</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()`

```
JAS_EXPORT int jas_stream_ungetc (
    jas\_stream\_t * stream,
    int c )
```

Put a character back on a stream.

Parameters

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

The character `c` (which was presumably read previously from the stream `stream`) is put back on the stream (as if it had not yet been read). In other words, this function undoes the effect of [jas_stream_getc\(\)](#). It is unspecified what happens

if the character put back was not the one originally read. The number of characters that can be pushed back onto the stream for subsequent reading is limited. Trying to push back too many characters on a stream will result in an error. The approximate limit is given by the value of `JAS_STREAM_MAXPUTBACK`.

Returns

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

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_rlimit()` 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_fmtnfo_t](#)
Image format information.

Macros

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

- `#define jas_image_cmpttlx(image, cmptno) ((image)->cmpts_[cmptno]->tlx_)`
Get the x-coordinate of the top-left corner of a component.
- `#define jas_image_cmpttly(image, cmptno) ((image)->cmpts_[cmptno]->tly_)`
Get the y-coordinate of the top-left corner of a component.
- `#define jas_image_cmptbrx(image, cmptno)`
Get the x-coordinate of the bottom-right corner of a component (plus "one").
- `#define jas_image_cmptbry(image, cmptno)`
Get the y-coordinate of the bottom-right corner of a component (plus "one").
- `#define jas_image_cmprofn(image) ((image)->cmprofn_)`
Get the color management profile of an image.
- `#define jas_image_setcmprofn(image, cmprofn) ((image)->cmprofn_ = cmprofn)`
Set the color management profile for an image.

Typedefs

- `typedef int_fast32_t jas_image_coord_t`
Image coordinate.
- `typedef int_fast16_t jas_image_colorspc_t`
Color space (e.g., RGB, YCbCr).
- `typedef int_fast32_t jas_image_cmpttype_t`
Component type (e.g., color, opacity).
- `typedef int_fast16_t jas_image_smpltype_t`
Component sample data format (e.g., real/integer, signedness, precision).

Functions

- `JAS_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_sampcmt` (`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_writecmt2` (`jas_image_t` *image, unsigned cmptno, `jas_image_coord_t` x, `jas_image_coord_t` y, `jas_image_coord_t` width, `jas_image_coord_t` height, const long *buf)
Write sample data in a component of an image.
- JAS_EXPORT int `jas_image_readcmt2` (`jas_image_t` *image, unsigned cmptno, `jas_image_coord_t` x, `jas_image_coord_t` y, `jas_image_coord_t` width, `jas_image_coord_t` height, long *buf)
Read sample data in a component of an image.
- JAS_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 )    ((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 )    ((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 ) ((image)->clrspc_)
```

Get the color model used by the image.

14.4.2.4 jas_image_cmprof

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

Get the height of a component.

14.4.2.8 `jas_image_cmptstep`

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

Get the width of a component.

14.4.2.15 `jas_image_height`

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

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

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

Get the number of image components.

14.4.2.18 `jas_image_setclrspc`

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

Set the color model for an image.

14.4.2.19 `jas_image_setcmprof`

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

Set the color management profile for an image.

14.4.2.20 `jas_image_tlx`

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

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

14.4.2.21 `jas_image_tly`

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

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

14.4.2.22 jas_image_width

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

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

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_smpltype_t

```
typedef int_fast16_t jas_image_smpltype_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_cmxform_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_CMXFOM_NUMINTENTS](#) 4
Number of rendering intents.
- #define [jas_clrspc_create](#)(fam, mbr) (((fam) << 8) | (mbr))
Create a color space.
- #define [jas_clrspc_fam](#)(clrspc) ((clrspc) >> 8)
Get the family of a color space.
- #define [jas_clrspc_mbr](#)(clrspc) ((clrspc) & 0xff)
Get the (family) member of a color space.
- #define [jas_clrspc_isgeneric](#)(clrspc) (![jas_clrspc_mbr](#)(clrspc))
Test if a color space is generic.
- #define [jas_clrspc_isunknown](#)(clrspc) ((clrspc) & JAS_CLRSPC_UNKNOWNMASK)
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) ((prof)->clrspc)
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 ) (((fam) << 8) | (mbr))
```

Create a color space.

14.5.2.2 `jas_clrspc_fam`

```
#define jas_clrspc_fam(  
    clrspc ) ((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 ) (!jas\_clrspc\_mbr(clrspc))
```

Test if a color space is generic.

14.5.2.6 `jas_clrspc_isunknown`

```
#define jas_clrspc_isunknown(  
    clrspc ) ((clrspc) & JAS_CLRSPC_UNKNOWNMASK)
```

Test if a color space is unknown.

14.5.2.7 `jas_clrspc_mbr`

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

```
typedef struct jas_cmpxform_s jas_cmpxform_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_iccattrtab_dump()`

```
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.

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_iccattrname_t name,
    jas_iccattrval_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_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_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_matind_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.
- 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_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_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()

```
static 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()

```
static 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()`

```
static 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()`

```
static 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()`

```
static 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()`

```
static 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()`

```
static 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()

```
static 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()

```
static 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()`

```
static 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()`

```
static void jas_matrix_set (  
    jas\_matrix\_t * matrix,  
    jas_matind_t i,  
    jas_matind_t j,  
    jas_seqent_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_seqent_t val )
```

Set all elements of a matrix to the specified value.

14.6.2.25 `jas_matrix_setv()`

```
static 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()`

```
static 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()`

```
static 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 * 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.

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()`

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

Destroy a 2-D sequence.

14.6.2.32 `jas_seq2d_empty()`

```
static 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()`

```
static 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()`

```
static 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()`

```
static 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()

```
static 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()

```
static 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()

```
static 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()

```
static 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()

```
static 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()

```
static 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()

```
static 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()

```
static 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()

```
static 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()

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

Destroy a 1-D sequence.

14.6.2.46 jas_seq_end()

```
static 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()

```
static 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()

```
static 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()

```
static 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()

```
static 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) JAS_INTTOFIX(fix_t, fracbits, 0)
• #define JAS_FIX_ONE(fix_t, fracbits) JAS_INTTOFIX(fix_t, fracbits, 1)
• #define JAS_FIX_HALF(fix_t, fracbits) (JAS_CAST(fix_t, 1) << ((fracbits) - 1))
• #define JAS_INTTOFIX(fix_t, fracbits, x) (JAS_CAST(fix_t, x) << (fracbits))
• #define JAS_FIXTOINT(fix_t, fracbits, x) JAS_CAST(int, (x) >> (fracbits))
• #define JAS_FIXTODBL(fix_t, fracbits, x) (JAS_CAST(double, x) / JAS_FIX_ONE(fix_t, fracbits))
• #define JAS_DBLTOFIX(fix_t, fracbits, x) JAS_CAST(fix_t, ((x) * JAS_CAST(double, JAS_FIX_ONE(fix_t, fracbits))))
• #define JAS_FIX_ADD JAS_FIX_ADD_FAST
• #define JAS_FIX_ADD_FAST(fix_t, fracbits, x, y) ((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) JAS_CAST(fix_t, ((x) * (y)))
• #define JAS_FIX_MULBYINT_OFLOW(fix_t, fracbits, x, y) JAS_FIX_MULBYINT_FAST(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) JAS_CAST(fix_t, (JAS_CAST(bigfix_t, x) << (fracbits)) / (y))
• #define JAS_FIX_DIV_UFLOW(fix_t, fracbits, bigfix_t, x, y) JAS_FIX_DIV_FAST(fix_t, fracbits, bigfix_t, x, y)
• #define JAS_FIX_NEG JAS_FIX_NEG_FAST
• #define JAS_FIX_NEG_FAST(fix_t, fracbits, x) ~(x)
• #define JAS_FIX_NEG_OFLOW(fix_t, fracbits, x) (((x) < 0) ? ~(x) > 0 || JAS_FIX_OFLOW(), ~(x)) : ~(x))
• #define JAS_FIX_ASF JAS_FIX_ASF_FAST
• #define JAS_FIX_ASF_FAST(fix_t, fracbits, x, n) ((x) << (n))
• #define JAS_FIX_ASF_OFLOW(fix_t, fracbits, x, n) (((x) << (n)) >> (n)) == (x) || JAS_FIX_OFLOW(), (x) << (n))
• #define JAS_FIX_ASR JAS_FIX_ASR_FAST
• #define JAS_FIX_ASR_FAST(fix_t, fracbits, x, n) ((x) >> (n))
• #define JAS_FIX_ASR_UFLOW(fix_t, fracbits, x, n) JAS_FIX_ASR_FAST(fix_t, fracbits, x, n)
• #define JAS_FIX_SUB(fix_t, fracbits, x, y) JAS_FIX_ADD(fix_t, fracbits, x, JAS_FIX_NEG(fix_t, fracbits, y))
• #define JAS_FIX_PLUSEQ(fix_t, fracbits, x, y) ((x) = JAS_FIX_ADD(fix_t, fracbits, x, y))
• #define JAS_FIX_MINUSEQ(fix_t, fracbits, x, y) ((x) = JAS_FIX_SUB(fix_t, fracbits, x, y))
• #define JAS_FIX_MULEQ(fix_t, fracbits, bigfix_t, x, y) ((x) = JAS_FIX_MUL(fix_t, fracbits, bigfix_t, x, y))
• #define JAS_FIX_ABS(fix_t, fracbits, x) (((x) >= 0) ? (x) : (JAS_FIX_NEG(fix_t, fracbits, x)))
• #define JAS_FIX_ISINT(fix_t, fracbits, x) (JAS_FIX_FLOOR(fix_t, fracbits, x) == (x))
• #define JAS_FIX_SGN(fix_t, fracbits, x) ((x) >= 0 ? 1 : (-1))
• #define JAS_FIX_CMP(fix_t, fracbits, x, y) ((x) > (y) ? 1 : (((x) == (y)) ? 0 : (-1)))
• #define JAS_FIX_LT(fix_t, fracbits, x, y) ((x) < (y))
• #define JAS_FIX_LTE(fix_t, fracbits, x, y) ((x) <= (y))
• #define JAS_FIX_GT(fix_t, fracbits, x, y) ((x) > (y))
• #define JAS_FIX_GTE(fix_t, fracbits, x, y) ((x) >= (y))
• #define JAS_FIX_ROUND(fix_t, fracbits, x)
• #define JAS_FIX_FLOOR(fix_t, fracbits, x) ((x) & ~(JAS_FIX_ONE(fix_t, fracbits) - 1))

```

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 )  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 )  (((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 )  ((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_ASF

```
#define JAS_FIX_ASF JAS_FIX_ASF_FAST
```

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

14.7.2.7 JAS_FIX_ASF_FAST

```
#define JAS_FIX_ASF_FAST(
    fix_t,
    fracbits,
    x,
    n ) ((x) << (n))
```

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

14.7.2.8 JAS_FIX_ASF_OFLOW

```
#define JAS_FIX_ASF_OFLOW(
    fix_t,
    fracbits,
    x,
    n ) (((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 )    ((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 )    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 )    ((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 )    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 )  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 )  ((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 )  ((x) > (y))
```

Greater than.

14.7.2.18 JAS_FIX_GTE

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

Greater than or equal.

14.7.2.19 JAS_FIX_HALF

```
#define JAS_FIX_HALF(
    fix_t,
    fracbits )  (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 )  (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 )  ((x) < (y))
```

Less than.

14.7.2.22 JAS_FIX_LTE

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

Less than or equal.

14.7.2.23 JAS_FIX_MINUSEQ

```
#define JAS_FIX_MINUSEQ(  
    fix_t,  
    fracbits,  
    x,  
    y )  ((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 ) 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 )  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 )  ((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 )  (-(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 )  (((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 )  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 )  ((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 )  ((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 )  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 )  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 )  (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 )  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 )  (JAS_CAST(fix_t, x) << (fracbits))
```

Convert an int to a fixed-point number.

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()

```
static int jas_logtype_getclass (
    jas_logtype_t type ) [inline], [static]
```

Get the class of a logtype.

14.8.4.2 jas_logtype_getpriority()

```
static int jas_logtype_getpriority (
    jas_logtype_t type ) [inline], [static]
```

Get the priority of a logtype.

14.8.4.3 jas_logtype_init()

```
static 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_nonnull\(\)](#)

```
JAS_ATTRIBUTE_PURE JAS_EXPORT const jas\_taginfo\_t * jas\_taginfo\_nonnull (
    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 )
```

Split a string into tokens based on specified delimiters.

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

```
1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
```

```

52  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60  *
61  * __END_OF_JASPER_LICENSE__
62  */
63
64  /*
65  * Windows Bitmap File Library
66  *
67  * $Id$
68  */
69
70  #ifndef BMP_COD_H
71  #define BMP_COD_H
72
73  /*****
74  * Includes.
75  *****/
76
77  #include "jasper/jas_types.h"
78
79  /*****
80  * Constants and macros.
81  *****/
82
83  #define BMP_MAGIC      0x4d42
84  /* The signature for a BMP file. */
85
86  #define BMP_HDRLLEN    14
87  /* The nominal header length. */
88
89  #define BMP_INFOLEN    40
90  /* The nominal info length. */
91
92  #define BMP_PALLEN(info) ((info)->numcolors * 4)
93  /* The length of the palette. */
94
95  #define BMP_HASPAL(info) ((info)->numcolors > 0)
96  /* Is this a palettized image? */
97
98  /* Encoding types. */
99  #define BMP_ENC_RGB      0 /* No special encoding. */
100 #define BMP_ENC_RLE8     1 /* Run length encoding. */
101 #define BMP_ENC_RLE4     2 /* Run length encoding. */
102
103 /*****
104 * Types.
105 *****/
106
107 /* BMP header. */
108 typedef struct {
109
110     int_fast16_t magic;
111     /* The signature (a.k.a. the magic number). */
112
113     int_fast32_t siz;
114     /* The size of the file in 32-bit words. */
115
116     int_fast16_t reserved1;
117     /* Ask Bill Gates what this is all about. */
118
119     int_fast16_t reserved2;
120     /* Ditto. */
121
122     int_fast32_t off;
123     /* The offset of the bitmap data from the bitmap file header in bytes. */
124
125 } bmp_hdr_t;
126
127 /* Palette entry. */
128 typedef struct {
129
130     int_fast16_t red;
131     /* The red component. */
132

```

```

133     int_fast16_t grn;
134     /* The green component. */
135
136     int_fast16_t blu;
137     /* The blue component. */
138
139     int_fast16_t res;
140     /* Reserved. */
141
142 } bmp_palent_t;
143
144 /* BMP info. */
145 typedef struct {
146
147     int_fast32_t len;
148     /* The length of the bitmap information header in bytes. */
149
150     int_fast32_t width;
151     /* The width of the bitmap in pixels. */
152
153     int_fast32_t height;
154     /* The height of the bitmap in pixels. */
155
156     int_fast8_t topdown;
157     /* The bitmap data is specified in top-down order. */
158
159     int_fast16_t numplanes;
160     /* The number of planes. This must be set to a value of one. */
161
162     int_fast16_t depth;
163     /* The number of bits per pixel. */
164
165     int_fast32_t enctype;
166     /* The type of compression used. */
167
168     int_fast32_t siz;
169     /* The size of the image in bytes. */
170
171     int_fast32_t hres;
172     /* The horizontal resolution in pixels/metre. */
173
174     int_fast32_t vres;
175     /* The vertical resolution in pixels/metre. */
176
177     int_fast32_t numcolors;
178     /* The number of color indices used by the bitmap. */
179
180     int_fast32_t mincolors;
181     /* The number of color indices important for displaying the bitmap. */
182
183     bmp_palent_t *palents;
184     /* The colors should be listed in order of importance. */
185
186 } bmp_info_t;
187
188 /*****
189  * Functions and macros.
190  *****/
191
192 #define bmp_issupported(hdr, info) \
193     ((hdr)->magic == BMP_MAGIC && !(hdr)->reserved1 && \
194      ! (hdr)->reserved2 && (info)->numplanes == 1 && \
195      ((info)->depth == 8 || (info)->depth == 24) && \
196      (info)->enctype == BMP_ENC_RGB)
197 /* Is this type of BMP file supported? */
198
199 #define bmp_haspal(info) \
200     ((info)->depth == 8)
201 /* Is there a palette? */
202
203 int bmp_numcmpts(bmp_info_t *info);
204 /* Get the number of components. */
205
206 bmp_info_t *bmp_info_create(void);
207 /* Create BMP information. */
208
209 void bmp_info_destroy(bmp_info_t *info);
210 /* Destroy BMP information. */
211
212 int bmp_isgrayscalepal(bmp_palent_t *palents, int numpalents);
213 /* Does the specified palette correspond to a grayscale image? */

```

```

214
215 #endif

```

16.2 bmp_enc.h

```

1 /*
2  * Copyright (c) 2002 Michael David Adams.
3  * All rights reserved.
4  */
5
6 /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37 * PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
43 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
48 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
50 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58 *
59 * __END_OF_JASPER_LICENSE__
60 */
61
62 #ifndef BMP_ENC_H
63 #define BMP_ENC_H
64
65 typedef struct {
66
67     int numcmpts;
68     int cmpts[4];
69
70 } bmp_enc_t;
71
72 #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_CMIFORM_NUMINTENTS](#) 4
Number of rendering intents.
- #define [jas_clrspc_create](#)(fam, mbr) (((fam) << 8) | (mbr))
Create a color space.
- #define [jas_clrspc_fam](#)(clrspc) ((clrspc) >> 8)
Get the family of a color space.
- #define [jas_clrspc_mbr](#)(clrspc) ((clrspc) & 0xff)
Get the (family) member of a color space.
- #define [jas_clrspc_isgeneric](#)(clrspc) (![jas_clrspc_mbr](#)(clrspc))
Test if a color space is generic.
- #define [jas_clrspc_isunknown](#)(clrspc) ((clrspc) & JAS_CLRSPC_UNKNOWNMASK)

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_cmpprof_clrspc](#)(prof) ((prof)->clrspc)

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_cmxfom_op_t](#)

Transform operations.

- enum [jas_cmxfom_intent_t](#)

Rendering intents.

- enum [jas_cmxfom_optm_t](#)

Transform optimization.

Functions

- JAS_EXPORT [jas_cmpprof_t](#) * [jas_cmpprof_createfromiccp](#)(const [jas_iccp_t](#) *iccp)

Create a color-management profile from an ICC profile.

- JAS_EXPORT [jas_cmpprof_t](#) * [jas_cmpprof_createfromclrspc](#)([jas_clrspc_t](#) clrspc)

Create a color-management profile from a color space.

- JAS_EXPORT void [jas_cmpprof_destroy](#)([jas_cmpprof_t](#) *prof)

Destroy a color-management profile.

- JAS_EXPORT [jas_cmpprof_t](#) * [jas_cmpprof_copy](#)(const [jas_cmpprof_t](#) *prof)

Copy a color-management profile.

- JAS_EXPORT [jas_iccp_t](#) * [jas_iccp_createfromcmp](#)(const [jas_cmpprof_t](#) *prof)

Create a ICC profile from a CM profile.

- JAS_EXPORT [jas_cmxfom_t](#) * [jas_cmxfom_create](#)(const [jas_cmpprof_t](#) *inprof, const [jas_cmpprof_t](#) *outprof, const [jas_cmpprof_t](#) *proofprof, [jas_cmxfom_op_t](#) op, [jas_cmxfom_intent_t](#) intent, [jas_cmxfom_optm_t](#) optimize)

Create a transform from a CM profile.

- JAS_EXPORT void [jas_cmxfom_destroy](#)([jas_cmxfom_t](#) *xfom)

Destroy a transform.

- JAS_EXPORT int [jas_cmxfom_apply](#)(const [jas_cmxfom_t](#) *xfom, 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.](#)

```

1 /*
2  * Copyright (c) 2002-2003 Michael David Adams.
3  * All rights reserved.
4  */
5
6 /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
43 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
48 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
50 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58 *
59 * __END_OF_JASPER_LICENSE__
60 */
61
62 #ifndef JAS_CM_H
63 #define JAS_CM_H
64
65 /*****\
66 */

```

```

72 \*****/
73
74 /* The configuration header file should be included first. */
75 #include <jasper/jas_config.h>
76
77 #include <jasper/jas_types.h>
78 #include <jasper/jas_icc.h>
79
80 #ifdef __cplusplus
81 extern "C" {
82 #endif
83
84 /*****\
85 * Types and Macros.
86 \*****/
87
88 typedef unsigned jas_clrspc_t;
89
90 typedef enum {
91     JAS_CMXFORM_OP_FWD = 0,
92     JAS_CMXFORM_OP_REV = 1,
93     JAS_CMXFORM_OP_PROOF = 2,
94     JAS_CMXFORM_OP_GAMUT = 3,
95 } jas_cmxform_op_t;
96
97 typedef enum {
98     JAS_CMXFORM_INTENT_PER = 0,
99     JAS_CMXFORM_INTENT_RELCLR = 1,
100    JAS_CMXFORM_INTENT_ABSCLR = 2,
101    JAS_CMXFORM_INTENT_SAT = 3,
102 } jas_cmxform_intent_t;
103
104 #define JAS_CMXFORM_NUMINTENTS 4
105
106 typedef enum {
107     JAS_CMXFORM_OPTM_SPEED = 0,
108     JAS_CMXFORM_OPTM_SIZE = 1,
109     JAS_CMXFORM_OPTM_ACC = 2,
110 } jas_cmxform_optm_t;
111
112 #define jas_clrspc_create(fam, mbr) (((fam) << 8) | (mbr))
113
114 #define jas_clrspc_fam(clrspc) ((clrspc) >> 8)
115
116 #define jas_clrspc_mbr(clrspc) ((clrspc) & 0xff)
117
118 #define jas_clrspc_isgeneric(clrspc) (!jas_clrspc_mbr(clrspc))
119
120 #define jas_clrspc_isunknown(clrspc) ((clrspc) & JAS_CLRSPC_UNKNOWNMASK)
121
122 #define JAS_CLRSPC_UNKNOWNMASK 0x4000
123
124 #define JAS_CLRSPC_FAM_UNKNOWN 0
125 #define JAS_CLRSPC_FAM_XYZ 1
126 #define JAS_CLRSPC_FAM_LAB 2
127 #define JAS_CLRSPC_FAM_GRAY 3
128 #define JAS_CLRSPC_FAM_RGB 4
129 #define JAS_CLRSPC_FAM_YCBCR 5
130
131 #define JAS_CLRSPC_UNKNOWN JAS_CLRSPC_UNKNOWNMASK
132 #define JAS_CLRSPC_CIEXYZ jas_clrspc_create(JAS_CLRSPC_FAM_XYZ, 1)
133 #define JAS_CLRSPC_CIELAB jas_clrspc_create(JAS_CLRSPC_FAM_LAB, 1)
134 #define JAS_CLRSPC_SGRAY jas_clrspc_create(JAS_CLRSPC_FAM_GRAY, 1)
135 #define JAS_CLRSPC_SRGB jas_clrspc_create(JAS_CLRSPC_FAM_RGB, 1)
136 #define JAS_CLRSPC_SYCBCR jas_clrspc_create(JAS_CLRSPC_FAM_YCBCR, 1)
137
138 #define JAS_CLRSPC_GENRGB jas_clrspc_create(JAS_CLRSPC_FAM_RGB, 0)
139 #define JAS_CLRSPC_GENGRAY jas_clrspc_create(JAS_CLRSPC_FAM_GRAY, 0)
140 #define JAS_CLRSPC_GENYCBCR jas_clrspc_create(JAS_CLRSPC_FAM_YCBCR, 0)
141
142 #define JAS_CLRSPC_CHANIND_YCBCR_Y 0
143 #define JAS_CLRSPC_CHANIND_YCBCR_CB 1
144 #define JAS_CLRSPC_CHANIND_YCBCR_CR 2
145
146 #define JAS_CLRSPC_CHANIND_RGB_R 0
147 #define JAS_CLRSPC_CHANIND_RGB_G 1
148 #define JAS_CLRSPC_CHANIND_RGB_B 2
149
150 #define JAS_CLRSPC_CHANIND_GRAY_Y 0
151
152 typedef double jas_cmreal_t;

```

```

211
212 struct jas_cmpxform_s;
213
214 typedef struct {
215     long *buf;
216     unsigned prec;
217     int sgnd;
218     unsigned width;
219     unsigned height;
220 } jas_cmcmptfmt_t;
221
222 typedef struct {
223     unsigned numcmpts;
224     jas_cmcmptfmt_t *cmptfmts;
225 } jas_cmpixmap_t;
226
227 typedef struct {
228     void (*destroy)(struct jas_cmpxform_s *pxform);
229     int (*apply)(const struct jas_cmpxform_s *pxform, const jas_cmreal_t *in, jas_cmreal_t *out,
230         unsigned cnt);
231     void (*dump)(struct jas_cmpxform_s *pxform);
232 } jas_cmpxformops_t;
233
234 typedef struct {
235     jas_cmreal_t *data;
236     unsigned size;
237 } jas_cmshapmatlut_t;
238
239 typedef struct {
240     int mono;
241     int order;
242     int useluts;
243     int usemat;
244     jas_cmshapmatlut_t luts[3];
245     jas_cmreal_t mat[3][4];
246 } jas_cmshapmat_t;
247
248 typedef struct {
249     int order;
250 } jas_cmshaplut_t;
251
252 typedef struct {
253     unsigned inclrspc;
254     unsigned outclrspc;
255 } jas_cmclrspcconv_t;
256
257 typedef struct jas_cmpxform_s {
258     unsigned refcnt;
259     const jas_cmpxformops_t *ops;
260     unsigned numinchans;
261     unsigned numoutchans;
262     union {
263         max_align_t dummy;
264         jas_cmshapmat_t shapmat;
265         jas_cmshaplut_t shaplut;
266         jas_cmclrspcconv_t clrspcconv;
267     } data;
268 } jas_cmpxform_t;
269
270 typedef struct {
271     unsigned numpxforms;
272     unsigned maxpxforms;
273     jas_cmpxform_t **pxforms;
274 } jas_cmpxformseq_t;
275
276 typedef struct {
277     unsigned numinchans;
278     unsigned numoutchans;
279     jas_cmpxformseq_t *pxformseq;
280 } jas_cmxform_t;
281
282 #define JAS_CMPROF_TYPE_DEV 1
283 #define JAS_CMPROF_TYPE_CLRSPC 2
284
285 #define JAS_CMPROF_NUMPXFORMSEQS 13
286
287 typedef struct {
288     jas_clrspc_t clrspc;
289     unsigned numchans;
290     unsigned refclrspc;
291     unsigned numrefchans;

```

```

334     jas_iccprof_t *iccprof;
335     jas_cmxformseq_t *pxformseqs[JAS_CMPROF_NUMPXFORMSEQS];
336 } jas_cmprof_t;
337
338 /*****
339 *
340 \*****/
341
342 #if 0
343 typedef int_fast32_t jas_cmattnname_t;
344 typedef int_fast32_t jas_cmattnval_t;
345 typedef int_fast32_t jas_cmattnrtype_t;
346 /* Load a profile. */
347 int jas_cmprof_load(jas_cmprof_t *prof, jas_stream_t *in, unsigned fmt);
348 /* Save a profile. */
349 int jas_cmprof_save(jas_cmprof_t *prof, jas_stream_t *out, unsigned fmt);
350 /* Set an attribute of a profile. */
351 int jas_cm_prof_setattr(jas_cm_prof_t *prof, jas_cm_attnname_t name, void *val);
352 /* Get an attribute of a profile. */
353 void *jas_cm_prof_getattr(jas_cm_prof_t *prof, jas_cm_attnname_t name);
354 #endif
355
356 /*****
357 * Color-management (CM) profile class.
358 \*****/
359
360 JAS_EXPORT
361 jas_cmprof_t *jas_cmprof_createfromiccprof(const jas_iccprof_t *iccprof);
362
363 JAS_EXPORT
364 jas_cmprof_t *jas_cmprof_createfromclrspc(jas_clrspc_t clrspc);
365
366 JAS_EXPORT
367 void jas_cmprof_destroy(jas_cmprof_t *prof);
368
369 JAS_EXPORT
370 jas_cmprof_t *jas_cmprof_copy(const jas_cmprof_t *prof);
371
372 JAS_EXPORT
373 jas_iccprof_t *jas_iccprof_createfromcmprof(const jas_cmprof_t *prof);
374
375 /*****
376 * Color-Management (CM) Transform.
377 \*****/
378
379 JAS_EXPORT
380 jas_cmxform_t *jas_cmxform_create(const jas_cmprof_t *inprof,
381     const jas_cmprof_t *outprof, const jas_cmprof_t *proofprof,
382     jas_cmxform_op_t op, jas_cmxform_intent_t intent,
383     jas_cmxform_optm_t optimize);
384
385 JAS_EXPORT
386 void jas_cmxform_destroy(jas_cmxform_t *xform);
387
388 JAS_EXPORT
389 int jas_cmxform_apply(const jas_cmxform_t *xform, const jas_cmpixmap_t *in,
390     jas_cmpixmap_t *out);
391
392 /*****
393 * Miscellany.
394 \*****/
395
396 unsigned jas_clrspc_numchans(jas_clrspc_t clrspc);
397
398 #define jas_cmprof_clrspc(prof) ((prof)->clrspc)
399
400 #ifdef __cplusplus
401 }
402 #endif
403 #endif

```

16.5 jas_compiler.h File Reference

Compiler-related macros.

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

Macros

- `#define JAS_UNUSED(x) ((void) 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 ) ((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.](#)

```
1 /* __START_OF_JASPER_LICENSE__
2  *
3  * JasPer License Version 2.0
4  *
5  * Copyright (c) 2001-2006 Michael David Adams
6  * Copyright (c) 1999-2000 Image Power, Inc.
7  * Copyright (c) 1999-2000 The University of British Columbia
8  *
9  * All rights reserved.
10 *
11 * Permission is hereby granted, free of charge, to any person (the
12 * "User") obtaining a copy of this software and associated documentation
13 * files (the "Software"), to deal in the Software without restriction,
14 * including without limitation the rights to use, copy, modify, merge,
15 * publish, distribute, and/or sell copies of the Software, and to permit
16 * persons to whom the Software is furnished to do so, subject to the
17 * following conditions:
18 *
19 * 1. The above copyright notices and this permission notice (which
20 * includes the disclaimer below) shall be included in all copies or
21 * substantial portions of the Software.
22 *
23 * 2. The name of a copyright holder shall not be used to endorse or
24 * promote products derived from the Software without specific prior
25 * written permission.
26 *
27 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
28 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
```

```

29 * THIS DISCLAIMER.  THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
30 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
31 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
32 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.  IN NO
33 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
34 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
35 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
36 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
37 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.  NO ASSURANCES ARE
38 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
39 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
40 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
41 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
42 * PROPERTY RIGHTS OR OTHERWISE.  AS A CONDITION TO EXERCISING THE RIGHTS
43 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
44 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY.  THE SOFTWARE
45 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
46 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
47 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
48 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
49 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
50 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
51 * RISK ACTIVITIES").  THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
52 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
53 *
54 * __END_OF_JASPER_LICENSE__
55 */
56
62 #ifndef JAS_COMPILER_H
63 #define JAS_COMPILER_H
64
65 /* The configuration header file should be included first. */
66 #include <jasper/jas_config.h>
67
68 #ifdef __MSC_VER
69 #   ifndef __cplusplus
70 #       undef inline
71 #       define inline __inline
72 #   endif
73 #endif
74
75 #if defined(__GNUC__)
76 #   define JAS_DEPRECATED __attribute__((deprecated))
77 #else
78 #   define JAS_DEPRECATED
79 #endif
80
81 #if defined(__GNUC__)
82 #   define JAS_ATTRIBUTE_CONST __attribute__((const))
83 #else
84 #   define JAS_ATTRIBUTE_CONST
85 #endif
86
87 #if defined(__GNUC__)
88 #   define JAS_ATTRIBUTE_PURE __attribute__((pure))
89 #else
90 #   define JAS_ATTRIBUTE_PURE
91 #endif
92
93 #if defined(__GNUC__)
94 #   define JAS_FORCE_INLINE inline __attribute__((always_inline))
95 #else
96 #   define JAS_FORCE_INLINE inline
97 #endif
98
99 #if defined(__GNUC__)
100 #   if __GNUC__ > 4 || (__GNUC__ == 4 && __GNUC_MINOR__ >= 5)
101 #       define JAS_UNREACHABLE() __builtin_unreachable()
102 #   else
103 #       define JAS_UNREACHABLE()
104 #   endif
105 #elif defined(__clang__)
106 #   define JAS_UNREACHABLE() __builtin_unreachable()
107 #elif defined(__MSC_VER)
108 #   define JAS_UNREACHABLE() __assume(0)
109 #else
110 #   define JAS_UNREACHABLE()
111 #endif
112
113 #if defined(__GNUC__)
114 #   define JAS_LIKELY(x) __builtin_expect (!!(x), 1)

```



```

115 #else
116 #     define JAS_LIKELY(x) (x)
117 #endif
118
119 #if defined(__GNUC__)
120 #     define JAS_UNLIKELY(x) __builtin_expect (!!(x), 0)
121 #else
122 #     define JAS_UNLIKELY(x) (x)
123 #endif
124
125 #if defined(__GNUC__) && __GNUC__ >= 6
126 #     define JAS_ATTRIBUTE_DISABLE_UBSAN \
127         __attribute__((no_sanitize_undefined))
128 #elif defined(__clang__)
129 #     define JAS_ATTRIBUTE_DISABLE_UBSAN \
130         __attribute__((no_sanitize("undefined")))
131 #else
132 #     define JAS_ATTRIBUTE_DISABLE_UBSAN
133 #endif
134
135 #ifdef __has_builtin
136 #define jas_has_builtin(x) __has_builtin(x)
137 #else
138 #define jas_has_builtin(x) 0
139 #endif
140
141 #define JAS_UNUSED(x) ((void) x)
142
143 #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) #x`
Convert to a string literal.
- `#define JAS_STRINGIFYX(x) JAS_STRINGIFY(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_loginfof](#) (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 ) #x
```

Convert to a string literal.

16.7.2.2 JAS_STRINGIFYX

```
#define JAS_STRINGIFYX(  
    x ) 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()

```
static 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.](#)

```
1 /*
2  * Copyright (c) 2001-2002 Michael David Adams.
3  * All rights reserved.
4  */
5
6 /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
43 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
```

```

44 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
48 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
50 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58 *
59 * __END_OF_JASPER_LICENSE__
60 */
61
62 #ifndef JAS_DEBUG_H
63 #define JAS_DEBUG_H
64
65 /*****
66 * Includes.
67 *****/
68
69 /* The configuration header file should be included first. */
70 #include <jasper/jas_config.h>
71
72 #include "jasper/jas_init.h"
73 #include "jasper/jas_debug.h"
74
75 #include <stdio.h>
76 #include <stdarg.h>
77
78 #ifdef __cplusplus
79 extern "C" {
80 #endif
81
82 /*****
83 * Macros and functions.
84 *****/
85
86 /* Output debugging information to standard error provided that the debug
87 level is set sufficiently high. */
88 #if !defined(NDEBUG)
89 #define JAS_DBGLOG(n, x) \
90     ((jas_get_debug_level() >= (n)) ? (jas_eprintf x) : 0)
91 #else
92 #define JAS_DBGLOG(n, x)
93 #endif
94
95 #if !defined(NDEBUG)
96 #define JAS_LOGDEBUGF(n, ...) \
97     ((jas_get_debug_level() >= (n)) ? jas_logdebugf((n), __VA_ARGS__) : 0)
98 #else
99 #define JAS_LOGDEBUGF(n, ...)
100 #endif
101
102 #endif
103
104 JAS_EXPORT
105 void jas_deprecated(const char *fmt, ...);
106
107 JAS_DEPRECATED
108 static inline
109 int jas_getdbglevel(void)
110 {
111     jas_deprecated("jas_getdbglevel is deprecated\n");
112     return jas_get_debug_level();
113 }
114
115 JAS_EXPORT
116 int jas_setdbglevel(int dbglevel);
117
118 JAS_EXPORT
119 int jas_eprintf(const char *fmt, ...);
120
121 JAS_EXPORT
122 int jas_logprintf(const char *fmt, ...);
123
124 JAS_EXPORT
125 int jas_logerrorf(const char *fmt, ...);
126

```

```

161 JAS_EXPORT
162 int jas_logwarnf(const char *fmt, ...);
163
164 JAS_EXPORT
165 int jas_loginf(const char *fmt, ...);
166
167 JAS_EXPORT
168 int jas_logdebugf(int priority, const char *fmt, ...);
169
170 int jas_logmemdump(const void *data, size_t len);
171
172 JAS_EXPORT
173 int jas_memdump(FILE *out, const void *data, size_t len);
174
175 #define JAS_STRINGIFY(x) #x
176
177 #define JAS_STRINGIFYX(x) JAS_STRINGIFY(x)
178
179 #ifdef __cplusplus
180 }
181 #endif
182 #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.](#)

```

1 /* __START_OF_JASPER_LICENSE__
2  *
3  * JasPer License Version 2.0
4  *
5  * Copyright (c) 2001-2006 Michael David Adams
6  * Copyright (c) 1999-2000 Image Power, Inc.
7  * Copyright (c) 1999-2000 The University of British Columbia
8  *
9  * All rights reserved.
10 *
11 * Permission is hereby granted, free of charge, to any person (the
12 * "User") obtaining a copy of this software and associated documentation
13 * files (the "Software"), to deal in the Software without restriction,
14 * including without limitation the rights to use, copy, modify, merge,
15 * publish, distribute, and/or sell copies of the Software, and to permit
16 * persons to whom the Software is furnished to do so, subject to the
17 * following conditions:
18 *
19 * 1. The above copyright notices and this permission notice (which
20 * includes the disclaimer below) shall be included in all copies or
21 * substantial portions of the Software.
22 *
23 * 2. The name of a copyright holder shall not be used to endorse or

```

```

24  * promote products derived from the Software without specific prior
25  * written permission.
26  *
27  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
28  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
29  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
30  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
31  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
32  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
33  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
34  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
35  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
36  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
37  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
38  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
39  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
40  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
41  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
42  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
43  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
44  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
45  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
46  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
47  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
48  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
49  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
50  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
51  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
52  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
53  *
54  * __END_OF_JASPER_LICENSE__
55  */
56
62 #ifndef JAS_DLL_H
63 #define JAS_DLL_H
64
65 /* The configuration header file should be included first. */
66 #include <jasper/jas_config.h>
67
68 #include <jasper/jas_export_cmake.h>
69
70 /* For backward compatibility only. */
71 #define JAS_DLLEXPORT JAS_EXPORT
72 /* For backward compatibility only. */
73 #define JAS_DLLLOCAL JAS_LOCAL
74
75 #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) JAS_INTTOFIX(fix_t, fracbits, 0)`
- `#define JAS_FIX_ONE(fix_t, fracbits) JAS_INTTOFIX(fix_t, fracbits, 1)`
- `#define JAS_FIX_HALF(fix_t, fracbits) (JAS_CAST(fix_t, 1) << ((fracbits) - 1))`
- `#define JAS_INTTOFIX(fix_t, fracbits, x) (JAS_CAST(fix_t, x) << (fracbits))`
- `#define JAS_FIXTOINT(fix_t, fracbits, x) JAS_CAST(int, (x) >> (fracbits))`
- `#define JAS_FIXTODBL(fix_t, fracbits, x) (JAS_CAST(double, x) / JAS_FIX_ONE(fix_t, fracbits))`

- `#define JAS_DBLTOFIX(fix_t, fracbits, x) JAS_CAST(fix_t, ((x) * JAS_CAST(double, JAS_FIX_ONE(fix_t, fracbits))))`
- `#define JAS_FIX_ADD JAS_FIX_ADD_FAST`
- `#define JAS_FIX_ADD_FAST(fix_t, fracbits, x, y) ((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) JAS_CAST(fix_t, ((x) * (y)))`
- `#define JAS_FIX_MULBYINT_OFLOW(fix_t, fracbits, x, y) JAS_FIX_MULBYINT_FAST(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) JAS_CAST(fix_t, (JAS_CAST(bigfix_t, x) << (fracbits)) / (y))`
- `#define JAS_FIX_DIV_UFLOW(fix_t, fracbits, bigfix_t, x, y) JAS_FIX_DIV_FAST(fix_t, fracbits, bigfix_t, x, y)`
- `#define JAS_FIX_NEG JAS_FIX_NEG_FAST`
- `#define JAS_FIX_NEG_FAST(fix_t, fracbits, x) (-x)`
- `#define JAS_FIX_NEG_OFLOW(fix_t, fracbits, x) (((x) < 0) ? (-x) > 0 || JAS_FIX_OFLOW(), -x) : (-x))`
- `#define JAS_FIX_AS_L JAS_FIX_AS_L_FAST`
- `#define JAS_FIX_AS_L_FAST(fix_t, fracbits, x, n) ((x) << (n))`
- `#define JAS_FIX_AS_L_OFLOW(fix_t, fracbits, x, n) (((x) << (n)) >> (n)) == (x) || JAS_FIX_OFLOW(), (x) << (n))`
- `#define JAS_FIX_AS_R JAS_FIX_AS_R_FAST`
- `#define JAS_FIX_AS_R_FAST(fix_t, fracbits, x, n) ((x) >> (n))`
- `#define JAS_FIX_AS_R_UFLOW(fix_t, fracbits, x, n) JAS_FIX_AS_R_FAST(fix_t, fracbits, x, n)`
- `#define JAS_FIX_SUB(fix_t, fracbits, x, y) JAS_FIX_ADD(fix_t, fracbits, x, JAS_FIX_NEG(fix_t, fracbits, y))`
- `#define JAS_FIX_PLUSEQ(fix_t, fracbits, x, y) ((x) = JAS_FIX_ADD(fix_t, fracbits, x, y))`
- `#define JAS_FIX_MINUSEQ(fix_t, fracbits, x, y) ((x) = JAS_FIX_SUB(fix_t, fracbits, x, y))`
- `#define JAS_FIX_MULEQ(fix_t, fracbits, bigfix_t, x, y) ((x) = JAS_FIX_MUL(fix_t, fracbits, bigfix_t, x, y))`
- `#define JAS_FIX_ABS(fix_t, fracbits, x) (((x) >= 0) ? (x) : (JAS_FIX_NEG(fix_t, fracbits, x)))`
- `#define JAS_FIX_ISINT(fix_t, fracbits, x) (JAS_FIX_FLOOR(fix_t, fracbits, x) == (x))`
- `#define JAS_FIX_SGN(fix_t, fracbits, x) ((x) >= 0 ? 1 : (-1))`
- `#define JAS_FIX_CMP(fix_t, fracbits, x, y) ((x) > (y) ? 1 : (((x) == (y)) ? 0 : (-1)))`
- `#define JAS_FIX_LT(fix_t, fracbits, x, y) ((x) < (y))`
- `#define JAS_FIX_LTE(fix_t, fracbits, x, y) ((x) <= (y))`
- `#define JAS_FIX_GT(fix_t, fracbits, x, y) ((x) > (y))`
- `#define JAS_FIX_GTE(fix_t, fracbits, x, y) ((x) >= (y))`
- `#define JAS_FIX_ROUND(fix_t, fracbits, x)`
- `#define JAS_FIX_FLOOR(fix_t, fracbits, x) ((x) & (~JAS_FIX_ONE(fix_t, fracbits) - 1))`

16.11.1 Detailed Description

JasPer Fixed-Point Number Class.

16.12 jas_fix.h

[Go to the documentation of this file.](#)

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 #ifndef JAS_FIX_H
65 #define JAS_FIX_H
66
67 /*****\
68 * Includes.
69 \*****/
70
71 /* The configuration header file should be included first. */
72 #include <jasper/jas_config.h> /* IWYU pragma: keep */
73
74 #include <jasper/jas_types.h>
75
76 #ifdef __cplusplus

```

```

82 extern "C" {
83 #endif
84
85 /*****
86  * Constants.
87  *****/
88 #define JAS_FIX_ZERO(fix_t, fracbits) \
89     JAS_INTTOFIX(fix_t, fracbits, 0)
90
91 #define JAS_FIX_ONE(fix_t, fracbits) \
92     JAS_INTTOFIX(fix_t, fracbits, 1)
93
94 #define JAS_FIX_HALF(fix_t, fracbits) \
95     (JAS_CAST(fix_t, 1) « ((fracbits) - 1))
96
97 /*****
98  * Conversion operations.
99  *****/
100 #define JAS_INTTOFIX(fix_t, fracbits, x) \
101     (JAS_CAST(fix_t, x) « (fracbits))
102
103 #define JAS_FIXTOINT(fix_t, fracbits, x) \
104     JAS_CAST(int, (x) » (fracbits))
105
106 #define JAS_FIXTODBL(fix_t, fracbits, x) \
107     (JAS_CAST(double, x) / JAS_FIX_ONE(fix_t, fracbits))
108
109 #define JAS_DBLTOFIX(fix_t, fracbits, x) \
110     JAS_CAST(fix_t, ((x) * JAS_CAST(double, JAS_FIX_ONE(fix_t, fracbits))))
111
112 /*****
113  * Basic arithmetic operations.
114  * All other arithmetic operations are synthesized from these basic operations.
115  * There are three macros for each type of arithmetic operation.
116  * One macro always performs overflow/underflow checking, one never performs
117  * overflow/underflow checking, and one is generic with its behavior
118  * depending on compile-time flags.
119  * Only the generic macros should be invoked directly by application code.
120  *****/
121 #if !defined(DEBUG_OVERFLOW)
122 #define JAS_FIX_ADD                JAS_FIX_ADD_FAST
123 #else
124 #define JAS_FIX_ADD                JAS_FIX_ADD_OFLOW
125 #endif
126
127 #define JAS_FIX_ADD_FAST(fix_t, fracbits, x, y) ((x) + (y))
128
129 #define JAS_FIX_ADD_OFLOW(fix_t, fracbits, x, y) \
130     ((x) >= 0) ? \
131     ((y) >= 0) ? ((x) + (y) >= 0 || JAS_FIX_OFLOW(), (x) + (y)) : \
132     ((x) + (y)) : \
133     ((y) >= 0) ? ((x) + (y)) : ((x) + (y) < 0 || JAS_FIX_OFLOW(), \
134     (x) + (y))
135
136 #if !defined(DEBUG_OVERFLOW)
137 #define JAS_FIX_MUL                JAS_FIX_MUL_FAST
138 #else
139 #define JAS_FIX_MUL                JAS_FIX_MUL_OFLOW
140 #endif
141
142 #define JAS_FIX_MUL_FAST(fix_t, fracbits, bigfix_t, x, y) \
143     JAS_CAST(fix_t, (JAS_CAST(bigfix_t, x) * JAS_CAST(bigfix_t, y)) » \
144     (fracbits))
145
146 #define JAS_FIX_MUL_OFLOW(fix_t, fracbits, bigfix_t, x, y) \
147     ((JAS_CAST(bigfix_t, x) * JAS_CAST(bigfix_t, y)) » (fracbits)) == \
148     JAS_CAST(fix_t, (JAS_CAST(bigfix_t, x) * JAS_CAST(bigfix_t, y)) » \
149     (fracbits))) ? \
150     JAS_CAST(fix_t, (JAS_CAST(bigfix_t, x) * JAS_CAST(bigfix_t, y)) » \
151     (fracbits)) : JAS_FIX_OFLOW()
152
153 #if !defined(DEBUG_OVERFLOW)
154 #define JAS_FIX_MULBYINT            JAS_FIX_MULBYINT_FAST
155 #else
156 #define JAS_FIX_MULBYINT            JAS_FIX_MULBYINT_OFLOW
157 #endif
158
159 #define JAS_FIX_MULBYINT_FAST(fix_t, fracbits, x, y) \

```

```

186     JAS_CAST(fix_t, ((x) * (y)))
187
190 #define JAS_FIX_MULBYINT_OFLOW(fix_t, fracbits, x, y) \
191     JAS_FIX_MULBYINT_FAST(fix_t, fracbits, x, y)
192
194 #if !defined(DEBUG_OVERFLOW)
195 #define JAS_FIX_DIV          JAS_FIX_DIV_FAST
196 #else
197 #define JAS_FIX_DIV          JAS_FIX_DIV_UFLOW
198 #endif
199
202 #define JAS_FIX_DIV_FAST(fix_t, fracbits, bigfix_t, x, y) \
203     JAS_CAST(fix_t, (JAS_CAST(bigfix_t, x) « (fracbits)) / (y))
204
207 #define JAS_FIX_DIV_UFLOW(fix_t, fracbits, bigfix_t, x, y) \
208     JAS_FIX_DIV_FAST(fix_t, fracbits, bigfix_t, x, y)
209
211 #if !defined(DEBUG_OVERFLOW)
212 #define JAS_FIX_NEG          JAS_FIX_NEG_FAST
213 #else
214 #define JAS_FIX_NEG          JAS_FIX_NEG_OFLOW
215 #endif
216
218 #define JAS_FIX_NEG_FAST(fix_t, fracbits, x) \
219     (-(x))
220
222 /* Yes, overflow is actually possible for two's complement representations,
223    although highly unlikely to occur. */
224 #define JAS_FIX_NEG_OFLOW(fix_t, fracbits, x) \
225     (((x) < 0) ? (-(x) > 0 || JAS_FIX_OFLOW(), -(x)) : (-(x)))
226
228 #if !defined(DEBUG_OVERFLOW)
229 #define JAS_FIX_AS_L        JAS_FIX_AS_L_FAST
230 #else
231 #define JAS_FIX_AS_L        JAS_FIX_AS_L_OFLOW
232 #endif
233
236 #define JAS_FIX_AS_L_FAST(fix_t, fracbits, x, n) \
237     ((x) « (n))
238
241 #define JAS_FIX_AS_L_OFLOW(fix_t, fracbits, x, n) \
242     (((x) « (n)) » (n)) == (x) || JAS_FIX_OFLOW(), (x) « (n))
243
245 #if !defined(DEBUG_OVERFLOW)
246 #define JAS_FIX_AS_R        JAS_FIX_AS_R_FAST
247 #else
248 #define JAS_FIX_AS_R        JAS_FIX_AS_R_UFLOW
249 #endif
250
253 #define JAS_FIX_AS_R_FAST(fix_t, fracbits, x, n) \
254     ((x) » (n))
255
258 #define JAS_FIX_AS_R_UFLOW(fix_t, fracbits, x, n) \
259     JAS_FIX_AS_R_FAST(fix_t, fracbits, x, n)
260
261 /*****\
262 * Other basic arithmetic operations.
263 \*****/
264
266 #define JAS_FIX_SUB(fix_t, fracbits, x, y) \
267     JAS_FIX_ADD(fix_t, fracbits, x, JAS_FIX_NEG(fix_t, fracbits, y))
268
270 #define JAS_FIX_PLUSEQ(fix_t, fracbits, x, y) \
271     ((x) = JAS_FIX_ADD(fix_t, fracbits, x, y))
272
274 #define JAS_FIX_MINUSEQ(fix_t, fracbits, x, y) \
275     ((x) = JAS_FIX_SUB(fix_t, fracbits, x, y))
276
278 #define JAS_FIX_MULEQ(fix_t, fracbits, bigfix_t, x, y) \
279     ((x) = JAS_FIX_MUL(fix_t, fracbits, bigfix_t, x, y))
280
281 /*****\
282 * Miscellaneous operations.
283 \*****/
284
286 #define JAS_FIX_ABS(fix_t, fracbits, x) \
287     (((x) >= 0) ? (x) : (JAS_FIX_NEG(fix_t, fracbits, x)))
288
290 #define JAS_FIX_ISINT(fix_t, fracbits, x) \
291     (JAS_FIX_FLOOR(fix_t, fracbits, x) == (x))
292

```

```

294 #define JAS_FIX_SGN(fix_t, fracbits, x) \
295     ((x) >= 0 ? 1 : (-1))
296
297 /*****
298  * Relational operations.
299  *****/
300
301 #define JAS_FIX_CMP(fix_t, fracbits, x, y) \
302     ((x) > (y) ? 1 : (((x) == (y)) ? 0 : (-1)))
303
304 #define JAS_FIX_LT(fix_t, fracbits, x, y) \
305     ((x) < (y))
306
307 #define JAS_FIX_LTE(fix_t, fracbits, x, y) \
308     ((x) <= (y))
309
310 #define JAS_FIX_GT(fix_t, fracbits, x, y) \
311     ((x) > (y))
312
313 #define JAS_FIX_GTE(fix_t, fracbits, x, y) \
314     ((x) >= (y))
315
316 /*****
317  * Rounding functions.
318  *****/
319
320 #define JAS_FIX_ROUND(fix_t, fracbits, x) \
321     (((x) < 0) ? JAS_FIX_FLOOR(fix_t, fracbits, JAS_FIX_ADD(fix_t, fracbits, \
322     (x), JAS_FIX_HALF(fix_t, fracbits))) : \
323     JAS_FIX_NEG(fix_t, fracbits, JAS_FIX_FLOOR(fix_t, fracbits, \
324     JAS_FIX_ADD(fix_t, fracbits, (-x), JAS_FIX_HALF(fix_t, fracbits)))))
325
326 #define JAS_FIX_FLOOR(fix_t, fracbits, x) \
327     ((x) & (~(JAS_FIX_ONE(fix_t, fracbits) - 1)))
328
329 /*****
330  * The below macros are for internal library use only. Do not invoke them
331  * directly in application code.
332  *****/
333
334 /* Handle overflow. */
335 #define JAS_FIX_OVERFLOW() \
336     jas_logerrorf("overflow error: file %s, line %d\n", __FILE__, __LINE__)
337
338 /* Handle underflow. */
339 #define JAS_FIX_UFLOW() \
340     jas_logerrorf("underflow error: file %s, line %d\n", __FILE__, __LINE__)
341
342 #ifdef __cplusplus
343 }
344 #endif
345 #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.](#)

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
```

```

29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 #ifndef JAS_GETOPT_H
65 #define JAS_GETOPT_H
66
67 #ifdef __cplusplus
68 extern "C" {
69 #endif
70
71 /* The configuration header file should be included first. */
72 #include <jasper/jas_config.h>
73
74 /*****\
75 * Constants.
76 \*****/
77
78 #define JAS_GETOPT_EOF (-1)
79 #define JAS_GETOPT_ERR '?'
80
81 #define JAS_OPT_HASARG 0x01 /* option has argument */
82
83 /*****\
84 * Types.
85 \*****/
86
87 typedef struct {
88     /* The unique identifier for this option. */
89     int id;
90
91     /* The name of this option. */
92     const char *name;
93
94     /* option flags. */
95     int flags;
96 } jas_opt_t;
97
98 /*****\
99 * External data.
100 \*****/
101
102 JAS_EXPORT
103 extern int jas_optind;
104
105 JAS_EXPORT
106 extern const char *jas_optarg;
107
108 #endif

```

```

139 JAS_EXPORT
140 extern int jas_opterr;
141
142 /*****
143  * Prototypes.
144  *****/
145
146 JAS_EXPORT
147 int jas_getopt(int argc, char **argv, const jas_opt_t *opts);
148
149 #ifdef __cplusplus
150 }
151 #endif
152
153 #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_iccatrrval_t](#) * [jas_iccprof_getattr](#) (const [jas_iccprof_t](#) *prof, [jas_iccatrrname_t](#) name)
Get an attribute of an ICC profile.
- JAS_EXPORT int [jas_iccprof_setattr](#) ([jas_iccprof_t](#) *prof, [jas_iccatrrname_t](#) name, [jas_iccatrrval_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_iccatrrval_destroy](#) ([jas_iccatrrval_t](#) *attrval)
Destroy an ICC profile attribute.
- JAS_EXPORT int [jas_iccatrrval_allowmodify](#) ([jas_iccatrrval_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.](#)

```

1 /*
2  * Copyright (c) 2002-2003 Michael David Adams.
3  * All rights reserved.
4  */
5
6 /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
43 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS

```

```

46  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
48  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
50  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58  *
59  * __END_OF_JASPER_LICENSE__
60  */
61
62 #ifndef JAS_ICC_H
63 #define JAS_ICC_H
64
65 /* The configuration header file should be included first. */
66 #include <jasper/jas_config.h>
67
68 #include <jasper/jas_types.h>
69 #include <jasper/jas_stream.h>
70
71 #include <stdio.h>
72
73 #ifdef __cplusplus
74 extern "C" {
75 #endif
76
77 /* Profile file signature. */
78 #define JAS_ICC_MAGIC 0x61637370
79
80 #define JAS_ICC_HDRLEN 128
81
82 /* Profile/device class signatures. */
83 #define JAS_ICC_CLAS_IN 0x73636e72 /* input device */
84 #define JAS_ICC_CLAS_DPY 0x646e7472 /* display device */
85 #define JAS_ICC_CLAS_OUT 0x70727472 /* output device */
86 #define JAS_ICC_CLAS_LNK 0x6c696e6b /* device link */
87 #define JAS_ICC_CLAS_CNV 0x73706163 /* color space conversion */
88 #define JAS_ICC_CLAS_ABS 0x61627374 /* abstract */
89 #define JAS_ICC_CLAS_NAM 0x6e64636c /* named color */
90
91 /* Color space signatures. */
92 #define JAS_ICC_COLORSPC_XYZ 0x58595a20 /* XYZ */
93 #define JAS_ICC_COLORSPC_LAB 0x4c616220 /* LAB */
94 #define JAS_ICC_COLORSPC_LUV 0x4c757620 /* LUV */
95 #define JAS_ICC_COLORSPC_YCBCR 0x59436272 /* YCbCr */
96 #define JAS_ICC_COLORSPC_YXY 0x59787920 /* Yxy */
97 #define JAS_ICC_COLORSPC_RGB 0x52474220 /* RGB */
98 #define JAS_ICC_COLORSPC_GRAY 0x47524159 /* Gray */
99 #define JAS_ICC_COLORSPC_HSV 0x48535620 /* HSV */
100 #define JAS_ICC_COLORSPC_HLS 0x484c5320 /* HLS */
101 #define JAS_ICC_COLORSPC_CMYK 0x434d594b /* CMYK */
102 #define JAS_ICC_COLORSPC_CMY 0x434d5920 /* CMY */
103 #define JAS_ICC_COLORSPC_2 0x32434c52 /* 2 channel color */
104 #define JAS_ICC_COLORSPC_3 0x33434c52 /* 3 channel color */
105 #define JAS_ICC_COLORSPC_4 0x34434c52 /* 4 channel color */
106 #define JAS_ICC_COLORSPC_5 0x35434c52 /* 5 channel color */
107 #define JAS_ICC_COLORSPC_6 0x36434c52 /* 6 channel color */
108 #define JAS_ICC_COLORSPC_7 0x37434c52 /* 7 channel color */
109 #define JAS_ICC_COLORSPC_8 0x38434c52 /* 8 channel color */
110 #define JAS_ICC_COLORSPC_9 0x39434c52 /* 9 channel color */
111 #define JAS_ICC_COLORSPC_10 0x41434c52 /* 10 channel color */
112 #define JAS_ICC_COLORSPC_11 0x42434c52 /* 11 channel color */
113 #define JAS_ICC_COLORSPC_12 0x43434c52 /* 12 channel color */
114 #define JAS_ICC_COLORSPC_13 0x44434c52 /* 13 channel color */
115 #define JAS_ICC_COLORSPC_14 0x45434c52 /* 14 channel color */
116 #define JAS_ICC_COLORSPC_15 0x46434c52 /* 15 channel color */
117
118 /* Profile connection color space (PCS) signatures. */
119 #define JAS_ICC_REF_COLORSPC_XYZ 0x58595a20 /* CIE XYZ */
120 #define JAS_ICC_REF_COLORSPC_LAB 0x4c616220 /* CIE Lab */
121
122 /* Primary platform signatures. */
123 #define JAS_ICC_PLATFORM_APPL 0x4150504c /* Apple Computer */
124 #define JAS_ICC_PLATFORM_MSFT 0x4d534654 /* Microsoft */
125 #define JAS_ICC_PLATFORM_SGI 0x53474920 /* Silicon Graphics */
126 #define JAS_ICC_PLATFORM_SUNW 0x53554e57 /* Sun Microsystems */

```

```
137 #define JAS_ICC_PLATFORM_TGNT    0x54474e54 /* Taligent */
138
139 /* Profile flags. */
140 #define JAS_ICC_FLAGS_EMBED      0x01 /* embedded */
141 #define JAS_ICC_FLAGS_NOSEP      0x02 /* no separate use */
142
143 /* Attributes. */
144 #define JAS_ICC_ATTR_TRANS      0x01 /* transparent */
145 #define JAS_ICC_ATTR_MATTE      0x02 /* matte */
146
147 /* Rendering intents. */
148 #define JAS_ICC_INTENT_PER      0 /* perceptual */
149 #define JAS_ICC_INTENT_REL      1 /* relative colorimetric */
150 #define JAS_ICC_INTENT_SAT      2 /* saturation */
151 #define JAS_ICC_INTENT_ABS      3 /* absolute colorimetric */
152
153 /* Tag signatures. */
154 #define JAS_ICC_TAG_ATOB0      0x41324230 /* */
155 #define JAS_ICC_TAG_ATOB1      0x41324231 /* */
156 #define JAS_ICC_TAG_ATOB2      0x41324232 /* */
157 #define JAS_ICC_TAG_BLUMATCOL  0x6258595a /* */
158 #define JAS_ICC_TAG_BLUTRC     0x62545243 /* */
159 #define JAS_ICC_TAG_BTOA0      0x42324130 /* */
160 #define JAS_ICC_TAG_BTOA1      0x42324131 /* */
161 #define JAS_ICC_TAG_BTOA2      0x42324132 /* */
162 #define JAS_ICC_TAG_CALTIME    0x63616c74 /* */
163 #define JAS_ICC_TAG_CHARTARGET 0x74617267 /* */
164 #define JAS_ICC_TAG_CPYRT      0x63707274 /* */
165 #define JAS_ICC_TAG_CRDINFO     0x63726469 /* */
166 #define JAS_ICC_TAG_DEVMKRDDESC 0x646d6e64 /* */
167 #define JAS_ICC_TAG_DEVMDLDESC 0x646d6464 /* */
168 #define JAS_ICC_TAG_DEVSET      0x64657673 /* */
169 #define JAS_ICC_TAG_GAMUT      0x67616d74 /* */
170 #define JAS_ICC_TAG_GRYTRC     0x6b545243 /* */
171 #define JAS_ICC_TAG_GRNMATCOL  0x6758595a /* */
172 #define JAS_ICC_TAG_GRNTRC     0x67545243 /* */
173 #define JAS_ICC_TAG_LUM        0x6c756d69 /* */
174 #define JAS_ICC_TAG_MEASURE     0x6d656173 /* */
175 #define JAS_ICC_TAG_MEDIABLKPT 0x626b7074 /* */
176 #define JAS_ICC_TAG_MEDIWHIPT  0x77747074 /* */
177 #define JAS_ICC_TAG_NAMCOLR     0x6e636f6c /* */
178 #define JAS_ICC_TAG_NAMCOLR2    0x6e636c32 /* */
179 #define JAS_ICC_TAG_OUTRESP     0x72657370 /* */
180 #define JAS_ICC_TAG_PREVIEW0    0x70726530 /* */
181 #define JAS_ICC_TAG_PREVIEW1    0x70726531 /* */
182 #define JAS_ICC_TAG_PREVIEW2    0x70726532 /* */
183 #define JAS_ICC_TAG_PROFDDESC   0x64657363 /* */
184 #define JAS_ICC_TAG_PROFSEQDESC 0x70736571 /* */
185 #define JAS_ICC_TAG_PSDCRD0     0x70736430 /* */
186 #define JAS_ICC_TAG_PSCRDD1     0x70736431 /* */
187 #define JAS_ICC_TAG_PSCRDD2     0x70736432 /* */
188 #define JAS_ICC_TAG_PSCRDD3     0x70736433 /* */
189 #define JAS_ICC_TAG_PS2CSA      0x70733273 /* */
190 #define JAS_ICC_TAG_PS2RENINTENT 0x70733269 /* */
191 #define JAS_ICC_TAG_REDMATCOL   0x7258595a /* */
192 #define JAS_ICC_TAG_REDTRC      0x72545243 /* */
193 #define JAS_ICC_TAG_SCRNGDES     0x73637264 /* */
194 #define JAS_ICC_TAG_SCRNG       0x7363726e /* */
195 #define JAS_ICC_TAG_TECH        0x74656368 /* */
196 #define JAS_ICC_TAG_UCRBG       0x62666420 /* */
197 #define JAS_ICC_TAG_VIEWCONDDDESC 0x76756564 /* */
198 #define JAS_ICC_TAG_VIEWCOND    0x76696577 /* */
199
200 /* Type signatures. */
201 #define JAS_ICC_TYPE_CRDINFO     0x63726469 /* CRD information */
202 #define JAS_ICC_TYPE_CURV        0x63757276 /* curve */
203 #define JAS_ICC_TYPE_DATA        0x64617461 /* data */
204 #define JAS_ICC_TYPE_TIME        0x6474696d /* date/time */
205 #define JAS_ICC_TYPE_DEVSET      0x64657673 /* device settings */
206 #define JAS_ICC_TYPE_LUT16       0x6d667432 /* */
207 #define JAS_ICC_TYPE_LUT8        0x6d667431 /* */
208 #define JAS_ICC_TYPE_MEASURE     0x6d656173 /* */
209 #define JAS_ICC_TYPE_NAMCOLR     0x6e636f6c /* */
210 #define JAS_ICC_TYPE_NAMCOLR2    0x6e636c32 /* */
211 #define JAS_ICC_TYPE_PROFSEQDESC 0x70736571 /* profile sequence description */
212 #define JAS_ICC_TYPE_RESPCURVSET16 0x72637332 /* response curve set 16 */
213 #define JAS_ICC_TYPE_SF32        0x73663332 /* signed 32-bit fixed-point */
214 #define JAS_ICC_TYPE_SCRNG       0x7363726e /* screening */
215 #define JAS_ICC_TYPE_SIG         0x73696720 /* signature */
216 #define JAS_ICC_TYPE_TXTDESC     0x64657363 /* text description */
217 #define JAS_ICC_TYPE_TXT         0x74657874 /* text */
```

```

218 #define JAS_ICC_TYPE_UF32          0x75663332 /* unsigned 32-bit fixed-point */
219 #define JAS_ICC_TYPE_UCRBG        0x62666420 /* */
220 #define JAS_ICC_TYPE_UI16         0x75693136 /* */
221 #define JAS_ICC_TYPE_UI32         0x75693332 /* */
222 #define JAS_ICC_TYPE_UI8          0x75693038 /* */
223 #define JAS_ICC_TYPE_UI64         0x75693634 /* */
224 #define JAS_ICC_TYPE_VIEWCOND     0x76696577 /* */
225 #define JAS_ICC_TYPE_XYZ          0x58595a20 /* XYZ */
226
227 typedef uint_fast8_t jas_iccuint8_t;
228 typedef uint_fast16_t jas_iccuint16_t;
229 typedef uint_fast32_t jas_iccuint32_t;
230 typedef int_fast32_t jas_iccsint32_t;
231 typedef int_fast32_t jas_iccs15fixed16_t;
232 typedef uint_fast32_t jas_iccul6fixed16_t;
233 typedef uint_fast64_t jas_iccuint64_t;
234 typedef uint_fast32_t jas_iccsig_t;
235
236 typedef jas_iccsig_t jas_icctagsig_t;
237 typedef jas_iccsig_t jas_icctagtype_t;
238 typedef jas_iccsig_t jas_iccattrname_t;
239
240 /* Date/time type. */
241 typedef struct {
242     jas_iccuint16_t year;
243     jas_iccuint16_t month;
244     jas_iccuint16_t day;
245     jas_iccuint16_t hour;
246     jas_iccuint16_t min;
247     jas_iccuint16_t sec;
248 } jas_icctime_t;
249
250 /* XYZ type. */
251 typedef struct {
252     jas_iccs15fixed16_t x;
253     jas_iccs15fixed16_t y;
254     jas_iccs15fixed16_t z;
255 } jas_iccxyz_t;
256
257 /* Curve type. */
258 typedef struct {
259     jas_iccuint32_t numents;
260     jas_iccuint16_t *ents;
261 } jas_icccurv_t;
262
263 /* Text description type. */
264 typedef struct {
265     jas_iccuint32_t asclen;
266     char *ascdata; /* ASCII invariant description */
267     jas_iccuint32_t uclangcode; /* Unicode language code */
268     jas_iccuint32_t uclen; /* Unicode localizable description count */
269     jas_uchar *ucdata; /* Unicode localizable description */
270     jas_iccuint16_t sccode; /* ScriptCode code */
271     jas_iccuint8_t maclen; /* Localizable Macintosh description count */
272     jas_uchar macdata[69]; /* Localizable Macintosh description */
273 } jas_iccxtxtdesc_t;
274
275 /* Text type. */
276 typedef struct {
277     char *string; /* ASCII character string */
278 } jas_iccxtxt_t;
279
280 typedef struct {
281     jas_iccuint8_t numinchans;
282     jas_iccuint8_t numoutchans;
283     jas_iccsint32_t e[3][3];
284     jas_iccuint8_t clutlen;
285     jas_iccuint8_t *clut;
286     jas_iccuint16_t numintabents;
287     jas_iccuint8_t **intabbuf;
288     jas_iccuint8_t *intabbuf;
289     jas_iccuint16_t numouttabents;
290     jas_iccuint8_t **outtabbuf;
291     jas_iccuint8_t *outtabbuf;
292 } jas_icclut8_t;
293
294 typedef struct {
295     jas_iccuint8_t numinchans;
296     jas_iccuint8_t numoutchans;
297     jas_iccsint32_t e[3][3];
298     jas_iccuint8_t clutlen;

```

```

299     jas_iccuint16_t *clut;
300     jas_iccuint16_t numintabents;
301     jas_iccuint16_t **intabs;
302     jas_iccuint16_t *intabsbuf;
303     jas_iccuint16_t numouttabents;
304     jas_iccuint16_t **outtabs;
305     jas_iccuint16_t *outtabsbuf;
306 } jas_icclut16_t;
307
308 struct jas_iccattrval_s;
309
310 typedef struct {
311     void (*destroy)(struct jas_iccattrval_s *);
312     int (*copy)(struct jas_iccattrval_s *, const struct jas_iccattrval_s *);
313     int (*input)(struct jas_iccattrval_s *, jas_stream_t *, unsigned);
314 #ifdef JAS_ENABLE_ENCODER
315     int (*output)(struct jas_iccattrval_s *, jas_stream_t *);
316 #endif
317     unsigned (*getsize)(const struct jas_iccattrval_s *);
318     void (*dump)(const struct jas_iccattrval_s *, FILE *);
319 } jas_iccattrvalops_t;
320
321 /* Attribute value type (type and value information). */
322 typedef struct jas_iccattrval_s {
323     unsigned refcnt; /* reference count */
324     jas_iccsig_t type; /* type */
325     const jas_iccattrvalops_t *ops; /* type-dependent operations */
326     union {
327         jas_iccxyz_t xyz;
328         jas_icccurv_t curv;
329         jas_iccxtxtdesc_t txtdesc;
330         jas_iccxtxt_t txt;
331         jas_icclut8_t lut8;
332         jas_icclut16_t lut16;
333     } data; /* value */
334 } jas_iccattrval_t;
335
336 /* Header type. */
337 typedef struct {
338     jas_iccuint32_t size; /* profile size */
339     jas_iccsig_t cmntype; /* CMM type signature */
340     jas_iccuint32_t version; /* profile version */
341     jas_iccsig_t clas; /* profile/device class signature */
342     jas_iccsig_t colorspace; /* color space of data */
343     jas_iccsig_t refcolorspace; /* profile connection space */
344     jas_icctime_t ctime; /* creation time */
345     jas_iccsig_t magic; /* profile file signature */
346     jas_iccsig_t platform; /* primary platform */
347     jas_iccuint32_t flags; /* profile flags */
348     jas_iccsig_t maker; /* device manufacturer signature */
349     jas_iccsig_t model; /* device model signature */
350     jas_iccuint64_t attr; /* device setup attributes */
351     jas_iccsig_t intent; /* rendering intent */
352     jas_iccxyz_t illum; /* illuminant */
353     jas_iccsig_t creator; /* profile creator signature */
354 } jas_icchdr_t;
355
356 typedef struct {
357     jas_iccsig_t name;
358     jas_iccattrval_t *val;
359 } jas_iccattr_t;
360
361 typedef struct {
362     unsigned numattrs;
363     unsigned maxattrs;
364     jas_iccattr_t *attrs;
365 } jas_iccattrtab_t;
366
367 typedef struct jas_icctagtabent_s {
368     jas_iccuint32_t tag;
369     jas_iccuint32_t off;
370     jas_iccuint32_t len;
371     void *data;
372     struct jas_icctagtabent_s *first;
373 } jas_icctagtabent_t;
374
375 typedef struct {
376     jas_iccuint32_t numents;
377     jas_icctagtabent_t *ents;
378 } jas_icctagtab_t;
379

```

```

380 /* ICC profile type. */
381 typedef struct {
382     jas_icchdr_t hdr;
383     jas_icctagtab_t tagtab;
384     jas_iccatrrtab_t attrtab;
385 } jas_iccprof_t;
386
387 typedef struct {
388     jas_iccuint32_t type;
389     jas_iccatrrvalops_t ops;
390 } jas_iccatrrvalinfo_t;
391
396 JAS_EXPORT
397 jas_iccprof_t *jas_iccprof_load(jas_stream_t *in);
398
403 JAS_EXPORT
404 int jas_iccprof_save(jas_iccprof_t *prof, jas_stream_t *out);
405
410 JAS_EXPORT
411 void jas_iccprof_destroy(jas_iccprof_t *prof);
412
417 JAS_ATTRIBUTE_PURE
418 JAS_EXPORT
419 jas_iccatrrval_t *jas_iccprof_getattr(const jas_iccprof_t *prof,
420     jas_iccatrrname_t name);
421
426 JAS_EXPORT
427 int jas_iccprof_setattr(jas_iccprof_t *prof, jas_iccatrrname_t name,
428     jas_iccatrrval_t *val);
429
435 JAS_EXPORT
436 void jas_iccprof_dump(const jas_iccprof_t *prof, FILE *out);
437
442 JAS_EXPORT
443 jas_iccprof_t *jas_iccprof_copy(const jas_iccprof_t *prof);
444
449 JAS_EXPORT
450 int jas_iccprof_gethdr(const jas_iccprof_t *prof, jas_icchdr_t *hdr);
451
456 JAS_EXPORT
457 int jas_iccprof_sethdr(jas_iccprof_t *prof, const jas_icchdr_t *hdr);
458
463 JAS_EXPORT
464 void jas_iccatrrval_destroy(jas_iccatrrval_t *attrval);
465
469 JAS_EXPORT
470 void jas_iccatrrval_dump(const jas_iccatrrval_t *attrval, FILE *out);
471
477 JAS_EXPORT
478 int jas_iccatrrval_allowmodify(jas_iccatrrval_t **attrval);
479
484 JAS_EXPORT
485 jas_iccatrrval_t *jas_iccatrrval_clone(jas_iccatrrval_t *attrval);
486
491 JAS_EXPORT
492 jas_iccatrrval_t *jas_iccatrrval_create(jas_iccuint32_t type);
493
499 JAS_EXPORT
500 void jas_iccatrrtab_dump(const jas_iccatrrtab_t *attrtab, FILE *out);
501
506 JAS_EXPORT
507 jas_iccprof_t *jas_iccprof_createfrombuf(const jas_uchar *buf, unsigned len);
508
513 JAS_EXPORT
514 jas_iccprof_t *jas_iccprof_createfromclrspc(unsigned clrspc);
515
516 JAS_EXPORT
517 extern const jas_uchar jas_iccprofdata_srgb[];
518
519 JAS_EXPORT
520 extern const unsigned jas_iccprofdata_srgblen;
521
522 JAS_EXPORT
523 extern const jas_uchar jas_iccprofdata_sgray[];
524
525 JAS_EXPORT
526 extern const unsigned jas_iccprofdata_sgraylen;
527
532 #ifdef __cplusplus
533 }
534 #endif

```

```
535
536 #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_fmtinfo_t](#)
Image format information.

Macros

- #define [JAS_IMAGE_MAXFMTS](#) 32
The maximum number of image data formats supported.
- #define [jas_image_width](#)(image) ((image)->brx_ - (image)->tlx_)
Get the width of the image in units of the image reference grid.
- #define [jas_image_height](#)(image) ((image)->bry_ - (image)->tly_)
Get the height of the image in units of the image reference grid.
- #define [jas_image_tlx](#)(image) ((image)->tlx_)
Get the x-coordinate of the top-left corner of the image bounding box on the reference grid.
- #define [jas_image_tly](#)(image) ((image)->tly_)
Get the y-coordinate of the top-left corner of the image bounding box on the reference grid.
- #define [jas_image_brx](#)(image) ((image)->brx_)
Get the x-coordinate of the bottom-right corner of the image bounding box on the reference grid (plus one).
- #define [jas_image_bry](#)(image) ((image)->bry_)
Get the y-coordinate of the bottom-right corner of the image bounding box on the reference grid (plus one).
- #define [jas_image_numcmpts](#)(image) ((image)->numcmpts_)

- Get the number of image components.*

 - #define `jas_image_clrspc`(image) ((image)->clrspc_)
- Get the color model used by the image.*

 - #define `jas_image_setclrspc`(image, clrspc) ((image)->clrspc_ = (clrspc))
- Set the color model for an image.*

 - #define `jas_image_cmptwidth`(image, cmptno) ((image)->cmpts_[cmptno]->width_)
- Get the width of a component.*

 - #define `jas_image_cmptheight`(image, cmptno) ((image)->cmpts_[cmptno]->height_)
- Get the height of a component.*

 - #define `jas_image_cmptsgnd`(image, cmptno) ((image)->cmpts_[cmptno]->sgnd_)
- Get the signedness of the sample data for a component.*

 - #define `jas_image_cmptprec`(image, cmptno) ((image)->cmpts_[cmptno]->prec_)
- Get the precision of the sample data for a component.*

 - #define `jas_image_cmptstep`(image, cmptno) ((image)->cmpts_[cmptno]->hstep_)
- Get the horizontal subsampling factor for a component.*

 - #define `jas_image_cmptvstep`(image, cmptno) ((image)->cmpts_[cmptno]->vstep_)
- Get the vertical subsampling factor for a component.*

 - #define `jas_image_cmpttlx`(image, cmptno) ((image)->cmpts_[cmptno]->tlx_)
- Get the x-coordinate of the top-left corner of a component.*

 - #define `jas_image_cmpttly`(image, cmptno) ((image)->cmpts_[cmptno]->tly_)
- Get the y-coordinate of the top-left corner of a component.*

 - #define `jas_image_cmptbrx`(image, cmptno)
- Get the x-coordinate of the bottom-right corner of a component (plus "one").*

 - #define `jas_image_cmptbry`(image, cmptno)
- Get the y-coordinate of the bottom-right corner of a component (plus "one").*

 - #define `jas_image_cmpprof`(image) ((image)->cmpprof_)
- Get the color management profile of an image.*

 - #define `jas_image_setcmpprof`(image, cmpprof) ((image)->cmpprof_ = cmpprof)
- 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_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_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_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_sampcmt2](#) ([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_writecmt2](#) ([jas_image_t](#) *image, unsigned cmptno, [jas_image_coord_t](#) x, [jas_image_coord_t](#) y, [jas_image_coord_t](#) width, [jas_image_coord_t](#) height, const long *buf)
Write sample data in a component of an image.
- JAS_EXPORT int [jas_image_readcmt2](#) ([jas_image_t](#) *image, unsigned cmptno, [jas_image_coord_t](#) x, [jas_image_coord_t](#) y, [jas_image_coord_t](#) width, [jas_image_coord_t](#) height, long *buf)
Read sample data in a component of an image.
- JAS_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).

16.17.1 Detailed Description

JasPer Image Class.

16.18 jas_image.h

[Go to the documentation of this file.](#)

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2003 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 #ifndef JAS_IMAGE_H
65 #define JAS_IMAGE_H
66
67 /*****\
68 * Includes.
69 \*****/
70
71 /* The configuration header file should be included first. */
72 #include <jasper/jas_config.h>
73
74 #include <jasper/jas_stream.h>
75 #include <jasper/jas_types.h>
76 #include <jasper/jas_seq.h> /* IWYU pragma: export */

```

```

82 #include <jasper/jas_cm.h> /* IWYU pragma: export */
83 #include <stdio.h>
84
85 #ifdef __cplusplus
86 extern "C" {
87 #endif
88
89 /*****
90  * Constants.
91  *****/
92
93 /*
94  * Miscellaneous constants.
95  */
96
97 /* Basic units */
98 #define JAS_IMAGE_KIBI (JAS_CAST(size_t, 1024))
99 #define JAS_IMAGE_MEBI (JAS_IMAGE_KIBI * JAS_IMAGE_KIBI)
100
101 /* The threshold at which image data is no longer stored in memory. */
102 #define JAS_IMAGE_INMEMTHRESH (256 * JAS_IMAGE_MEBI)
103
104 /*
105  * Component types
106  */
107
108 #define JAS_IMAGE_CT_UNKNOWN 0x10000
109 #define JAS_IMAGE_CT_COLOR(n) ((n) & 0x7fff)
110 #define JAS_IMAGE_CT_OPACITY 0x08000
111
112 #define JAS_IMAGE_CT_RGB_R 0
113 #define JAS_IMAGE_CT_RGB_G 1
114 #define JAS_IMAGE_CT_RGB_B 2
115
116 #define JAS_IMAGE_CT_YCBCR_Y 0
117 #define JAS_IMAGE_CT_YCBCR_CB 1
118 #define JAS_IMAGE_CT_YCBCR_CR 2
119
120 #define JAS_IMAGE_CT_GRAY_Y 0
121
122 /*****
123  * Simple types.
124  *****/
125
126 typedef int_fast32_t jas_image_coord_t;
127 #define JAS_IMAGE_COORD_MAX INT_FAST32_MAX
128 #define JAS_IMAGE_COORD_MIN INT_FAST32_MIN
129
130 typedef int_fast16_t jas_image_colorspc_t;
131
132 typedef int_fast32_t jas_image_cmpttype_t;
133
134 typedef int_fast16_t jas_image_smpltype_t;
135
136 /*****
137  * Image class and supporting classes.
138  *****/
139
140 typedef struct {
141     /* The x-coordinate of the top-left corner of the component. */
142     jas_image_coord_t tlx_;
143
144     /* The y-coordinate of the top-left corner of the component. */
145     jas_image_coord_t tly_;
146
147     /* The horizontal sampling period in units of the reference grid. */
148     jas_image_coord_t hstep_;
149
150     /* The vertical sampling period in units of the reference grid. */
151     jas_image_coord_t vstep_;
152
153     /* The component width in samples. */
154     jas_image_coord_t width_;
155
156     /* The component height in samples. */
157     jas_image_coord_t height_;
158
159     /* The precision of the sample data (i.e., the number of bits per sample).
160      * If the samples are signed values, this quantity includes the sign bit. */
161     unsigned prec_;

```

```

191
192     /* The signedness of the sample data. */
193     int sgnd_;
194
195     /* The stream containing the component data. */
196     jas_stream_t *stream_;
197
198     /* The number of characters per sample in the stream. */
199     unsigned cps_;
200
201     /* The type of component (e.g., opacity, red, green, blue, luma). */
202     jas_image_cmpttype_t type_;
203
204 } jas_image_cmpt_t;
205
206 typedef struct {
207
208     /* The x-coordinate of the top-left corner of the image bounding box. */
209     jas_image_coord_t tlx_;
210
211     /* The y-coordinate of the top-left corner of the image bounding box. */
212     jas_image_coord_t tly_;
213
214     /* The x-coordinate of the bottom-right corner of the image bounding
215        box (plus one). */
216     jas_image_coord_t brx_;
217
218     /* The y-coordinate of the bottom-right corner of the image bounding
219        box (plus one). */
220     jas_image_coord_t bry_;
221
222     /* The number of components. */
223     unsigned numcmpts_;
224
225     /* The maximum number of components that this image can have (i.e., the
226        allocated size of the components array). */
227     unsigned maxcmpts_;
228
229     /* Per-component information. */
230     jas_image_cmpt_t **cmpts_;
231
232     /* The color space. */
233     jas_clrspc_t clrspc_;
234
235     /* The CM profile. */
236     jas_cmprof_t *cmprof_;
237
238     //bool inmem_;
239
240 } jas_image_t;
241
242 typedef struct {
243
244     /* The x-coordinate of the top-left corner of the component. */
245     jas_image_coord_t tlx_;
246
247     /* The y-coordinate of the top-left corner of the component. */
248     jas_image_coord_t tly_;
249
250     /* The horizontal sampling period in units of the reference grid. */
251     jas_image_coord_t hstep;
252
253     /* The vertical sampling period in units of the reference grid. */
254     jas_image_coord_t vstep;
255
256     /* The width of the component in samples. */
257     jas_image_coord_t width;
258
259     /* The height of the component in samples. */
260     jas_image_coord_t height;
261
262     /* The precision of the component sample data. */
263     unsigned prec;
264
265     /* The signedness of the component sample data. */
266     int sgnd;
267
268 } jas_image_cmptparm_t;
269
270 /*****
271  * File format related classes.

```

```

288 \*****/
289
293 #define JAS_IMAGE_MAXFMTS      32
294
298 typedef struct {
299
301     jas_image_t *(*decode)(jas_stream_t *in, const char *opts);
302
304     int (*encode)(jas_image_t *image, jas_stream_t *out, const char *opts);
305
307     int (*validate)(jas_stream_t *in);
308
309 } jas_image_fmtops_t;
310
314 typedef struct {
315
317     int id;
318
320     char *name;
321
322     /* The primary file name extension associated with this format. */
323     /* This member only exists for backward compatibility. */
324     char *ext;
325
327     char **exts;
328     size_t max_exts;
329     size_t num_exts;
330
332     int enabled;
333
335     char *desc;
336
338     jas_image_fmtops_t ops;
339
340 } jas_image_fmtinfo_t;
341
342 \*****/
343 * Image operations.
344 \*****/
345
349 JAS_EXPORT
350 jas_image_t *jas_image_create(unsigned numcmpts,
351     const jas_image_cmptparm_t *cmptparms, jas_clrspc_t clrspc);
352
356 JAS_EXPORT
357 jas_image_t *jas_image_create0(void);
358
362 JAS_EXPORT
363 jas_image_t *jas_image_copy(jas_image_t *image);
364
368 JAS_EXPORT
369 void jas_image_destroy(jas_image_t *image);
370
374 #define jas_image_width(image) \
375     ((image)->brx_ - (image)->tlx_)
376
380 #define jas_image_height(image) \
381     ((image)->bry_ - (image)->tly_)
382
387 #define jas_image_tlx(image) \
388     ((image)->tlx_)
389
394 #define jas_image_tly(image) \
395     ((image)->tly_)
396
401 #define jas_image_brx(image) \
402     ((image)->brx_)
403
408 #define jas_image_bry(image) \
409     ((image)->bry_)
410
414 #define jas_image_numcmpts(image) \
415     ((image)->numcmpts_)
416
420 #define jas_image_clrspc(image) \
421     ((image)->clrspc_)
422
426 #define jas_image_setclrspc(image, clrspc) \
427     ((image)->clrspc_ = (clrspc))
428
429 #define jas_image_cmpttype(image, cmptno) \

```

```

430     ((image)->cmpts_[cmptno]->type_)
431 #define jas_image_setcmpttype(image, cmptno, type) \
432     ((image)->cmpts_[cmptno]->type_ = (type))
433
437 #define jas_image_cmptwidth(image, cmptno) \
438     ((image)->cmpts_[cmptno]->width_)
439
443 #define jas_image_cmptheight(image, cmptno) \
444     ((image)->cmpts_[cmptno]->height_)
445
449 #define jas_image_cmptsgnd(image, cmptno) \
450     ((image)->cmpts_[cmptno]->sgnd_)
451
455 #define jas_image_cmptprec(image, cmptno) \
456     ((image)->cmpts_[cmptno]->prec_)
457
461 #define jas_image_cmptstep(image, cmptno) \
462     ((image)->cmpts_[cmptno]->hstep_)
463
467 #define jas_image_cmptvstep(image, cmptno) \
468     ((image)->cmpts_[cmptno]->vstep_)
469
473 #define jas_image_cmpttlx(image, cmptno) \
474     ((image)->cmpts_[cmptno]->tlx_)
475
479 #define jas_image_cmpttly(image, cmptno) \
480     ((image)->cmpts_[cmptno]->tly_)
481
486 #define jas_image_cmptbrx(image, cmptno) \
487     ((image)->cmpts_[cmptno]->tlx_ + (image)->cmpts_[cmptno]->width_ * \
488     (image)->cmpts_[cmptno]->hstep_)
489
494 #define jas_image_cmptbry(image, cmptno) \
495     ((image)->cmpts_[cmptno]->tly_ + (image)->cmpts_[cmptno]->height_ * \
496     (image)->cmpts_[cmptno]->vstep_)
497
501 JAS_ATTRIBUTE_PURE
502 JAS_EXPORT
503 bool jas_image_cmpt_domains_same(const jas_image_t *image);
504
509 JAS_ATTRIBUTE_PURE
510 JAS_EXPORT
511 uint_fast32_t jas_image_rawsize(const jas_image_t *image);
512
516 JAS_EXPORT
517 jas_image_t *jas_image_decode(jas_stream_t *in, int fmt, const char *optstr);
518
522 JAS_EXPORT
523 int jas_image_encode(jas_image_t *image, jas_stream_t *out, int fmt,
524     const char *optstr);
525
533 JAS_EXPORT
534 int jas_image_readcmpt(jas_image_t *image, unsigned cmptno,
535     jas_image_coord_t x, jas_image_coord_t y, jas_image_coord_t width,
536     jas_image_coord_t height, jas_matrix_t *data);
537
541 JAS_EXPORT
542 int jas_image_writecmpt(jas_image_t *image, unsigned cmptno,
543     jas_image_coord_t x, jas_image_coord_t y, jas_image_coord_t width,
544     jas_image_coord_t height, const jas_matrix_t *data);
545
549 JAS_EXPORT
550 void jas_image_delcmpt(jas_image_t *image, unsigned cmptno);
551
555 JAS_EXPORT
556 int jas_image_addcmpt(jas_image_t *image, int cmptno,
557     const jas_image_cmptparm_t *cmptparm);
558
562 JAS_EXPORT
563 int jas_image_copycmpt(jas_image_t *dstimage, unsigned dstcmptno,
564     jas_image_t *srcimage, unsigned srccmptno);
565
566 JAS_ATTRIBUTE_CONST
567 static inline bool JAS_IMAGE_CDT_GETSGND(uint_least8_t dtype)
568 {
569     return (dtype > 7) & 1;
570 }
571
572 JAS_ATTRIBUTE_CONST
573 static inline uint_least8_t JAS_IMAGE_CDT_SETSGND(bool sgnd)
574 {

```

```

575         return (uint_least8_t)sgnd < 7;
576     }
577
578     JAS_ATTRIBUTE_CONST
579     static inline uint_least8_t JAS_IMAGE_CDT_GETPREC(uint_least8_t dtype)
580     {
581         return dtype & 0x7f;
582     }
583
584     JAS_ATTRIBUTE_CONST
585     static inline uint_least8_t JAS_IMAGE_CDT_SETPREC(uint_least8_t dtype)
586     {
587         return dtype & 0x7f;
588     }
589
590     JAS_ATTRIBUTE_PURE
591     static inline uint_least8_t jas_image_cmptdtype(const jas_image_t *image,
592         unsigned cmptno)
593     {
594         return JAS_IMAGE_CDT_SETSGND(image->cmpts_[cmptno]->sgnd_) |
595             JAS_IMAGE_CDT_SETPREC(image->cmpts_[cmptno]->prec_);
596     }
597
601     JAS_EXPORT
602     int jas_image_depalettize(jas_image_t *image, unsigned cmptno,
603         unsigned numlutents, const int_fast32_t *lutents, unsigned dtype,
604         unsigned newcmptno);
605
609     JAS_EXPORT
610     int jas_image_readcmptsample(jas_image_t *image, unsigned cmptno, unsigned x,
611         unsigned y);
612
616     JAS_EXPORT
617     void jas_image_writecmptsample(jas_image_t *image, unsigned cmptno,
618         unsigned x, unsigned y, int_fast32_t v);
619
623     JAS_ATTRIBUTE_PURE
624     JAS_EXPORT
625     int jas_image_getcmptbytype(const jas_image_t *image, jas_image_cmpttype_t ctype);
626
627     /*****
628     * Image format-related operations.
629     *****/
630
634     JAS_EXPORT
635     void jas_image_clearfmts(void);
636
637     #if defined(JAS_FOR_INTERNAL_USE_ONLY)
638     void jas_image_clearfmts_internal(jas_image_fmtinfo_t *image_fmtinfos,
639         size_t *image_numfmts);
640     #endif
641
645     JAS_EXPORT
646     const jas_image_fmtinfo_t *jas_image_getfmtbyind(int index);
647
651     JAS_EXPORT
652     int jas_image_getnumfmts(void);
653
654     #if 0
655     JAS_EXPORT
656     int jas_image_delfmtbyid(int id);
657     #endif
658
666     JAS_EXPORT
667     int jas_image_setfmtenable(int index, int enabled);
668
669     #if 0
670     // TODO: should this be added?
671     JAS_EXPORT
672     int jas_image_getfmtindbyname(const char* name);
673     #endif
674
678     JAS_EXPORT
679     int jas_image_addfmt(int id, const char *name, const char *ext,
680         const char *desc, const jas_image_fmtops_t *ops);
681
682     #if defined(JAS_FOR_INTERNAL_USE_ONLY)
683     int jas_image_addfmt_internal(jas_image_fmtinfo_t *image_fmtinfos,
684         size_t *image_numfmts, int id, const char *name, const char *ext,
685         const char *desc, const jas_image_fmtops_t *ops);
686     #endif

```



```

687
691 JAS_ATTRIBUTE_PURE
692 JAS_EXPORT
693 int jas_image_strtofmt(const char *s);
694
698 JAS_ATTRIBUTE_CONST
699 JAS_EXPORT
700 const char *jas_image_fmtostr(int fmt);
701
705 JAS_ATTRIBUTE_CONST
706 JAS_EXPORT
707 const jas_image_fmtinfo_t *jas_image_lookupfmtbyid(int id);
708
712 JAS_ATTRIBUTE_PURE
713 JAS_EXPORT
714 const jas_image_fmtinfo_t *jas_image_lookupfmtbyname(const char *name);
715
719 JAS_ATTRIBUTE_PURE
720 JAS_EXPORT
721 int jas_image_fmtfromname(const char *filename);
722
729 JAS_ATTRIBUTE_PURE
730 JAS_EXPORT
731 int jas_image_getfmt(jas_stream_t *in);
732
736 #define jas_image_cmprof(image) ((image)->cmprof_)
737
742 JAS_ATTRIBUTE_PURE
743 JAS_EXPORT
744 int jas_image_ishomosamp(const jas_image_t *image);
745
749 JAS_EXPORT
750 int jas_image_sampcmpt(jas_image_t *image, unsigned cmptno, unsigned newcmptno,
751     jas_image_coord_t ho, jas_image_coord_t vo, jas_image_coord_t hs,
752     jas_image_coord_t vs, int sgnd, unsigned prec);
753
757 JAS_EXPORT
758 int jas_image_writecmpt2(jas_image_t *image, unsigned cmptno,
759     jas_image_coord_t x, jas_image_coord_t y, jas_image_coord_t width,
760     jas_image_coord_t height, const long *buf);
761
765 JAS_EXPORT
766 int jas_image_readcmpt2(jas_image_t *image, unsigned cmptno,
767     jas_image_coord_t x, jas_image_coord_t y, jas_image_coord_t width,
768     jas_image_coord_t height, long *buf);
769
773 #define jas_image_setcmprof(image, cmprof) ((image)->cmprof_ = cmprof)
774
778 JAS_EXPORT
779 jas_image_t *jas_image_chclrspc(jas_image_t *image,
780     const jas_cmprof_t *outprof, jas_cmxform_intent_t intent);
781
785 JAS_EXPORT
786 int jas_image_dump(jas_image_t *image, FILE *out);
787
788 /*****
789 * Image format-dependent operations.
790 *****/
791
792 #if defined(JAS_INCLUDE_JPG_CODEC)
793 /* Format-dependent operations for JPG support. */
794 //JAS_EXPORT
795 jas_image_t *jpg_decode(jas_stream_t *in, const char *optstr);
796 //JAS_EXPORT
797 int jpg_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
798 //JAS_EXPORT
799 int jpg_validate(jas_stream_t *in);
800 #endif
801
802 #if defined(JAS_INCLUDE_HEIC_CODEC)
803 /* Format-dependent operations for HEIC support. */
804 //JAS_EXPORT
805 jas_image_t *jas_heic_decode(jas_stream_t *in, const char *optstr);
806 //JAS_EXPORT
807 int jas_heic_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
808 //JAS_EXPORT
809 int jas_heic_validate(jas_stream_t *in);
810 #endif
811
812 #if defined(JAS_INCLUDE_MIF_CODEC)
813 /* Format-dependent operations for MIF support. */

```

```

814 //JAS_EXPORT
815 jas_image_t *mif_decode(jas_stream_t *in, const char *optstr);
816 //JAS_EXPORT
817 int mif_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
818 //JAS_EXPORT
819 int mif_validate(jas_stream_t *in);
820 #endif
821
822 #if defined(JAS_INCLUDE_PNM_CODEC)
823 /* Format-dependent operations for PNM support. */
824 //JAS_EXPORT
825 jas_image_t *pnm_decode(jas_stream_t *in, const char *optstr);
826 //JAS_EXPORT
827 int pnm_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
828 //JAS_EXPORT
829 int pnm_validate(jas_stream_t *in);
830 #endif
831
832 #if defined(JAS_INCLUDE_RAS_CODEC)
833 /* Format-dependent operations for Sun Rasterfile support. */
834 //JAS_EXPORT
835 jas_image_t *ras_decode(jas_stream_t *in, const char *optstr);
836 //JAS_EXPORT
837 int ras_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
838 //JAS_EXPORT
839 int ras_validate(jas_stream_t *in);
840 #endif
841
842 #if defined(JAS_INCLUDE_BMP_CODEC)
843 /* Format-dependent operations for BMP support. */
844 //JAS_EXPORT
845 jas_image_t *bmp_decode(jas_stream_t *in, const char *optstr);
846 //JAS_EXPORT
847 int bmp_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
848 //JAS_EXPORT
849 int bmp_validate(jas_stream_t *in);
850 #endif
851
852 #if defined(JAS_INCLUDE_JP2_CODEC)
853 /* Format-dependent operations for JP2 support. */
854 //JAS_EXPORT
855 jas_image_t *jp2_decode(jas_stream_t *in, const char *optstr);
856 //JAS_EXPORT
857 int jp2_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
858 //JAS_EXPORT
859 int jp2_validate(jas_stream_t *in);
860 #endif
861
862 #if defined(JAS_INCLUDE_JPC_CODEC)
863 /* Format-dependent operations for JPEG-2000 code stream support. */
864 //JAS_EXPORT
865 jas_image_t *jpc_decode(jas_stream_t *in, const char *optstr);
866 //JAS_EXPORT
867 int jpc_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
868 //JAS_EXPORT
869 int jpc_validate(jas_stream_t *in);
870 #endif
871
872 #if defined(JAS_INCLUDE_PGX_CODEC)
873 /* Format-dependent operations for PGX support. */
874 //JAS_EXPORT
875 jas_image_t *pgx_decode(jas_stream_t *in, const char *optstr);
876 //JAS_EXPORT
877 int pgx_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
878 //JAS_EXPORT
879 int pgx_validate(jas_stream_t *in);
880 #endif
881
882 #ifdef __cplusplus
883 }
884 #endif
885
886 #endif
887
888 #endif
889
890 #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.](#)

```

1 /*
2  * Copyright (c) 2001-2002 Michael David Adams.
3  * All rights reserved.
4  */
5
6 /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING

```

```

36  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.  IN NO
38  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.  NO ASSURANCES ARE
43  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47  * PROPERTY RIGHTS OR OTHERWISE.  AS A CONDITION TO EXERCISING THE RIGHTS
48  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY.  THE SOFTWARE
50  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56  * RISK ACTIVITIES").  THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58  *
59  * __END_OF_JASPER_LICENSE__
60  */
61
62 #ifndef JAS_INIT_H
63 #define JAS_INIT_H
64
65 /* The configuration header file should be included first. */
66 #include <jasper/jas_config.h>
67
68 #include "jasper/jas_malloc.h"
69 #include "jasper/jas_image.h"
70 #include "jasper/jas_log.h"
71
72 #include <stdarg.h>
73
74 #ifdef __cplusplus
75 extern "C" {
76 #endif
77
78 #if defined(JAS_FOR_INTERNAL_USE_ONLY)
79 /*
80  * Per-thread library context.
81  */
82 typedef struct {
83
84     /*
85      * The level of debugging checks/output enabled by the library.
86      * A larger value corresponds to a greater level of debugging checks/output.
87      */
88     int debug_level;
89
90     /*
91      * The function used to output error/warning/informational messages.
92      * int (*vlogmsgf)(jas_logtype_t type, const char *format, va_list ap);
93      */
94     jas_vlogmsgf_t *vlogmsgf;
95
96     /*
97      * The image format information to be used to populate the image format
98      * table.
99      */
100    size_t image_numfmts;
101    jas_image_fmtinfo_t image_fmtinfos[JAS_IMAGE_MAXFMTS];
102
103    /* The maximum number of samples allowable in an image to be decoded. */
104    size_t dec_default_max_samples;
105
106 } jas_ctx_t;
107 #endif
108
109 typedef void *jas_context_t;
110
111 /*****
112  * Library Run-Time Configuration.
113  *****/
114
115 JAS_EXPORT
116 void jas_conf_clear(void);

```

```

145
151 JAS_EXPORT
152 void jas_conf_set_multithread(int multithread);
153
163 JAS_EXPORT
164 void jas_conf_set_allocator(jas_allocator_t *allocator);
165
171 JAS_EXPORT
172 void jas_conf_set_debug_level(int debug_level);
173
187 JAS_EXPORT
188 void jas_conf_set_max_mem_usage(size_t max_mem);
189
197 JAS_EXPORT
198 void jas_conf_set_dec_default_max_samples(size_t max_samples);
199
207 JAS_EXPORT
208 void jas_conf_set_vlogmsgf(jas_vlogmsgf_t *func);
209
210 /*****
211  * Library Initialization and Cleanup.
212  *****/
213
232 JAS_EXPORT
233 int jas_init_library(void);
234
247 JAS_EXPORT
248 int jas_cleanup_library(void);
249
250 /*****
251  * Thread Initialization and Cleanup.
252  *****/
253
267 JAS_EXPORT
268 int jas_init_thread(void);
269
276 JAS_EXPORT
277 int jas_cleanup_thread(void);
278
279 /*****
280  * Legacy Initialization and Cleanup Functions.
281  *****/
282
309 JAS_EXPORT
310 int jas_init(void);
311
326 JAS_EXPORT
327 void jas_cleanup(void);
328
329 /*****
330  * Context Management
331  *****/
332
339 JAS_EXPORT
340 jas_context_t jas_context_create(void);
341
349 JAS_EXPORT
350 void jas_context_destroy(jas_context_t context);
351
358 JAS_EXPORT
359 jas_context_t jas_get_default_context(void);
360
367 JAS_EXPORT
368 jas_context_t jas_get_context(void);
369
376 JAS_EXPORT
377 void jas_set_context(jas_context_t context);
378
379 /*****
380  * Getting/Setting Context Properties
381  *****/
382
383 /* This function is only for internal use by the library. */
384 JAS_EXPORT
385 int jas_get_debug_level_internal(void);
386
387 /* This function is only for internal use by the library. */
388 JAS_EXPORT
389 size_t jas_get_dec_default_max_samples_internal(void);
390
391 /* This function is only for internal use by the library. */

```

```

392 JAS_EXPORT
393 jas_vlogmsggf_t *jas_get_vlogmsggf_internal(void);
394
395 #if defined(JAS_FOR_INTERNAL_USE_ONLY)
396 #if defined(JAS_HAVE_THREAD_LOCAL)
397 extern _Thread_local jas_ctx_t *jas_cur_ctx;
398 #endif
399
400 /* This function is only for internal use by the library. */
401 jas_ctx_t *jas_get_ctx_internal(void);
402
403 /* This function is only for internal use by the library. */
404 static inline jas_ctx_t *jas_get_ctx(void)
405 {
406     #if defined(JAS_HAVE_THREAD_LOCAL)
407         return jas_cur_ctx ? jas_cur_ctx : jas_get_ctx_internal();
408     #else
409         return JAS_CAST(jas_ctx_t *, jas_get_ctx_internal());
410     #endif
411 }
412 #endif
413
414 JAS_EXPORT
415 void jas_set_debug_level(int debug_level);
416
417 static inline int jas_get_debug_level(void)
418 {
419     #if defined(JAS_FOR_INTERNAL_USE_ONLY)
420         jas_ctx_t *ctx = jas_get_ctx();
421         return ctx->debug_level;
422     #else
423         return jas_get_debug_level_internal();
424     #endif
425 }
426
427 JAS_EXPORT
428 void jas_set_dec_default_max_samples(size_t max_samples);
429
430 static inline size_t jas_get_dec_default_max_samples(void)
431 {
432     #if defined(JAS_FOR_INTERNAL_USE_ONLY)
433         jas_ctx_t *ctx = jas_get_ctx();
434         return ctx->dec_default_max_samples;
435     #else
436         return jas_get_dec_default_max_samples_internal();
437     #endif
438 }
439
440 JAS_EXPORT
441 void jas_set_vlogmsggf(jas_vlogmsggf_t *func);
442
443 static inline
444 jas_vlogmsggf_t *jas_get_vlogmsggf(void)
445 {
446     #if defined(JAS_FOR_INTERNAL_USE_ONLY)
447         jas_ctx_t *ctx = jas_get_ctx();
448         return ctx->_vlogmsggf;
449     #else
450         return jas_get_vlogmsggf_internal();
451     #endif
452 }
453
454 #ifdef __cplusplus
455 }
456 #endif
457 #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.](#)

```

1 /*
2  * Copyright (c) 2001-2002 Michael David Adams.
3  * All rights reserved.
4  */
5
6 /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
43 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
48 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
50 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58 *
59 * __END_OF_JASPER_LICENSE__
60 */
61
62 #ifndef JAS_LOG_H
63 #define JAS_LOG_H
64
65 /******\
66 * Includes.
67 \*****/
68
69 /* The configuration header file should be included first. */
70 #include <jasper/jas_config.h>
71
72 #include <stdio.h>
73 #include <stdarg.h>
74
75 #ifdef __cplusplus
76 extern "C" {

```

```

82 #endif
83
89 /*****\
90 * Macros and functions.
91 \*****/
92
94 #define JAS_LOGTYPE_CLASS_NULL 0
96 #define JAS_LOGTYPE_CLASS_ERROR 1
98 #define JAS_LOGTYPE_CLASS_WARN 2
100 #define JAS_LOGTYPE_CLASS_INFO 3
102 #define JAS_LOGTYPE_CLASS_DEBUG 4
103 #define JAS_LOGTYPE_NUM_CLASSES 5
104
105 #define JAS_LOGTYPE_MAX_PRIORITY 16384
106
107 // NOTE: without the @struct, jas_logtype_t autolinks are not generated
112 typedef unsigned int jas_logtype_t;
113
117 typedef int (jas_vlogmsgf_t)(jas_logtype_t, const char *, va_list);
118
122 static inline jas_logtype_t jas_logtype_init(int clas, int priority)
123 {
124     assert(clas >= 0 && clas < JAS_LOGTYPE_NUM_CLASSES);
125     assert(priority >= 0 && priority <= JAS_LOGTYPE_MAX_PRIORITY);
126     return (clas & 0xf) | (priority << 4);
127 }
128
132 static inline int jas_logtype_getclass(jas_logtype_t type)
133 {
134     return type & 0xf;
135 }
136
140 static inline int jas_logtype_getpriority(jas_logtype_t type)
141 {
142     return type >> 4;
143 }
144
148 JAS_EXPORT
149 int jas_vlogmsgf(jas_logtype_t type, const char *fmt, va_list ap);
150
154 JAS_EXPORT
155 int jas_vlogmsgf_stderr(jas_logtype_t type, const char *fmt, va_list ap);
156
160 JAS_EXPORT
161 int jas_vlogmsgf_discard(jas_logtype_t type, const char *fmt, va_list ap);
162
167 #ifdef __cplusplus
168 }
169 #endif
170
171 #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.](#)

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64
65 #ifndef JAS_MALLOC_H
66 #define JAS_MALLOC_H
67
68 /* *****\
69  * Includes.
70  * *****/
71
72 /* The configuration header file should be included first. */
73 #include <jasper/jas_config.h>
74
75 #include <jasper/jas_types.h>
76 #include <jasper/jas_thread.h>

```

```

82
83 #include <stdio.h>
84
85 #ifdef __cplusplus
86 extern "C" {
87 #endif
88
89 /******\
90 * Types.
91 \*****/
92
101 typedef struct jas_allocator_s {
102
103     void (*cleanup)(struct jas_allocator_s *allocator);
104
105     void *(*alloc)(struct jas_allocator_s *allocator, size_t size);
106
107     void (*free)(struct jas_allocator_s *allocator, void *pointer);
108
109     void *(*realloc)(struct jas_allocator_s *allocator, void *pointer,
110                     size_t new_size);
111
112     void (*reserved[4])(void);
113 } jas_allocator_t;
114
115 typedef struct {
116
117     /* The base class. */
118     jas_allocator_t base;
119
120 } jas_std_allocator_t;
121
122 #if defined(JAS_FOR_INTERNAL_USE_ONLY)
123 /*
124 The allocator wrapper type.
125 This type is an allocator that adds memory usage tracking to another
126 allocator.
127 The allocator wrapper does not directly perform memory allocation itself.
128 Instead, it delegate to another allocator.
129 */
130 typedef struct {
131
132     /* The base class. */
133     jas_allocator_t base;
134
135     /* The delegated-to allocator. */
136     jas_allocator_t *delegate;
137
138     /* The maximum amount of memory that can be used by the allocator. */
139     size_t max_mem;
140
141     /* The current amount of memory in use by the allocator. */
142     size_t mem;
143
144 #if defined(JAS_THREADS)
145     /* A mutex for synchronized access to the allocator. */
146     jas_mutex_t mutex;
147 #endif
148 } jas_basic_allocator_t;
149 #endif
150
151 /******\
152 * Data.
153 \*****/
154
155 #if defined(JAS_FOR_INTERNAL_USE_ONLY)
156 extern jas_allocator_t *jas_allocator;
157 extern jas_std_allocator_t jas_std_allocator;
158 extern jas_basic_allocator_t jas_basic_allocator;
159 #endif
160
161 /******\
162 * Functions.
163 \*****/
164
165 JAS_EXPORT
166 void *jas_malloc(size_t size);
167
168 JAS_EXPORT

```

```

214 void jas_free(void *ptr);
215
224 JAS_EXPORT
225 void *jas_realloc(void *ptr, size_t size);
226
235 JAS_EXPORT
236 void *jas_calloc(size_t num_elements, size_t element_size);
237
242 JAS_EXPORT
243 void *jas_alloc2(size_t num_elements, size_t element_size);
244
249 JAS_EXPORT
250 void *jas_alloc3(size_t num_arrays, size_t array_size, size_t element_size);
251
256 JAS_EXPORT
257 void *jas_realloc2(void *ptr, size_t num_elements, size_t element_size);
258
280 JAS_EXPORT
281 void jas_set_max_mem_usage(size_t max_mem);
282
295 JAS_EXPORT
296 size_t jas_get_mem_usage(void);
297
311 JAS_EXPORT
312 void jas_std_allocator_init(jas_std_allocator_t *allocator);
313
323 JAS_EXPORT
324 void jas_allocator_cleanup(jas_allocator_t *allocator);
325
326 #if defined(JAS_FOR_INTERNAL_USE_ONLY)
327
328 /* This function is for internal library use only. */
329 void jas_set_allocator(jas_allocator_t* allocator);
330
331 /* This function is for internal library use only. */
332 void jas_basic_allocator_init(jas_basic_allocator_t *allocator,
333     jas_allocator_t *delegate, size_t max_mem);
334
335 #endif
336
352 JAS_EXPORT
353 size_t jas_get_total_mem_size(void);
354
359 #ifdef __cplusplus
360 }
361 #endif
362
363 #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.](#)

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 #ifndef JAS_MATH_H
65 #define JAS_MATH_H
66
67 /*****\
68 * Includes
69 \*****/
70
71 /* The configuration header file should be included first. */
72 #include <jasper/jas_config.h>
73
74 #include <jasper/jas_compiler.h>
75 #include <jasper/jas_types.h>
76
77

```

```

82 #include <assert.h>
83 #include <string.h>
84 #include <stdint.h>
85 #include <limits.h>
86
87 #ifdef __cplusplus
88 extern "C" {
89 #endif
90
91 /*****
92  * Macros
93  *****/
94
95 #define JAS_KIBI      JAS_CAST(size_t, 1024)
96 #define JAS_MEBI      (JAS_KIBI * JAS_KIBI)
97
98 /* Compute the absolute value. */
99 #define JAS_ABS(x) \
100     ((x) >= 0) ? (x) : (-(x))
101
102 /* Compute the minimum of two values. */
103 #define JAS_MIN(x, y) \
104     ((x) < (y)) ? (x) : (y)
105
106 /* Compute the maximum of two values. */
107 #define JAS_MAX(x, y) \
108     ((x) > (y)) ? (x) : (y)
109
110 /* Compute the remainder from division (where division is defined such
111    that the remainder is always nonnegative). */
112 #define JAS_MOD(x, y) \
113     (((x) < 0) ? (((-x) % (y)) ? ((y) - ((-x) % (y))) : (0)) : ((x) % (y)))
114
115 /* Compute the integer with the specified number of least significant bits
116    set to one. */
117 #define JAS_ONES(n) \
118     ((1 « (n)) - 1)
119
120 /*****
121  *
122  *****/
123
124 #if defined(__clang__) || (defined(__GNUC__) && __GNUC__ > 6)
125 /* suppress clang warning "shifting a negative signed value is
126    undefined" in the assertions below */
127 #pragma GCC diagnostic push
128 #pragma GCC diagnostic ignored "-Wshift-negative-value"
129 #endif
130
131 JAS_ATTRIBUTE_CONST
132 JAS_ATTRIBUTE_DISABLE_UBSAN
133 inline static int jas_int_asr(int x, unsigned n)
134 {
135     // Ensure that the shift of a negative value appears to behave as a
136     // signed arithmetic shift.
137     assert(((-1) » 1) == -1);
138     // The behavior is undefined when x is negative.
139     // We tacitly assume the behavior is equivalent to a signed
140     // arithmetic right shift.
141     return x » n;
142 }
143
144 JAS_ATTRIBUTE_CONST
145 JAS_ATTRIBUTE_DISABLE_UBSAN
146 inline static int jas_int_asl(int x, unsigned n)
147 {
148     // Ensure that the shift of a negative value appears to behave as a
149     // signed arithmetic shift.
150     assert(((-1) « 1) == -2);
151     // The behavior is undefined when x is negative.
152     // We tacitly assume the behavior is equivalent to a signed
153     // arithmetic left shift.
154     return x « n;
155 }
156
157 JAS_ATTRIBUTE_CONST
158 JAS_ATTRIBUTE_DISABLE_UBSAN
159 inline static int_least32_t jas_least32_asr(int_least32_t x, unsigned n)
160 {
161     // Ensure that the shift of a negative value appears to behave as a
162     // signed arithmetic shift.

```



```

163     assert(((JAS_CAST(int_least32_t, -1)) > 1) == JAS_CAST(int_least32_t, -1));
164     // The behavior is undefined when x is negative.
165     // We tacitly assume the behavior is equivalent to a signed
166     // arithmetic right shift.
167     return x >> n;
168 }
169
170 JAS_ATTRIBUTE_CONST
171 JAS_ATTRIBUTE_DISABLE_UBSAN
172 inline static int_least32_t jas_least32_asl(int_least32_t x, unsigned n)
173 {
174     // Ensure that the shift of a negative value appears to behave as a
175     // signed arithmetic shift.
176     assert(((JAS_CAST(int_least32_t, -1)) < 1) == JAS_CAST(int_least32_t, -2));
177     // The behavior is undefined when x is negative.
178     // We tacitly assume the behavior is equivalent to a signed
179     // arithmetic left shift.
180     return x << n;
181 }
182
183 JAS_ATTRIBUTE_CONST
184 JAS_ATTRIBUTE_DISABLE_UBSAN
185 inline static int_fast32_t jas_fast32_asr(int_fast32_t x, unsigned n)
186 {
187     // Ensure that the shift of a negative value appears to behave as a
188     // signed arithmetic shift.
189     assert(((JAS_CAST(int_fast32_t, -1)) > 1) == JAS_CAST(int_fast32_t, -1));
190     // The behavior is undefined when x is negative.
191     // We tacitly assume the behavior is equivalent to a signed
192     // arithmetic right shift.
193     return x >> n;
194 }
195
196 JAS_ATTRIBUTE_CONST
197 JAS_ATTRIBUTE_DISABLE_UBSAN
198 inline static int_fast32_t jas_fast32_asl(int_fast32_t x, unsigned n)
199 {
200     // Ensure that the shift of a negative value appears to behave as a
201     // signed arithmetic shift.
202     assert(((JAS_CAST(int_fast32_t, -1)) < 1) == JAS_CAST(int_fast32_t, -2));
203     // The behavior is undefined when x is negative.
204     // We tacitly assume the behavior is equivalent to a signed
205     // arithmetic left shift.
206     return x << n;
207 }
208
209 #if defined(__clang__) || (defined(__GNUC__) && __GNUC__ > 6)
210 #pragma GCC diagnostic pop
211 #endif
212
213 /*****\
214 * Safe integer arithmetic (i.e., with overflow checking).
215 \*****/
216
217 /* Compute the product of two size_t integers with overflow checking. */
218 inline static bool jas_safe_size_mul(size_t x, size_t y, size_t *result)
219 {
220     #if jas_has_builtin(__builtin_mul_overflow) || (defined(__GNUC__) && __GNUC__ > 5)
221         size_t result_buffer;
222         bool valid = !__builtin_mul_overflow(x, y, &result_buffer);
223         if (valid && result) {
224             *result = result_buffer;
225         }
226         return valid;
227     #else
228         /* Check if overflow would occur */
229         if (x && y > SIZE_MAX / x) {
230             /* Overflow would occur. */
231             return false;
232         }
233         if (result) {
234             *result = x * y;
235         }
236         return true;
237     #endif
238 }
239
240 /* Compute the product of three size_t integers with overflow checking. */
241 inline static bool jas_safe_size_mul3(size_t a, size_t b, size_t c,
242     size_t *result)
243 {

```

```

244     size_t tmp;
245     if (!jas_safe_size_mul(a, b, &tmp) ||
246         !jas_safe_size_mul(tmp, c, &tmp)) {
247         return false;
248     }
249     if (result) {
250         *result = tmp;
251     }
252     return true;
253 }
254
255 /* Compute the sum of two size_t integers with overflow checking. */
256 inline static bool jas_safe_size_add(size_t x, size_t y, size_t *result)
257 {
258     #if jas_has_builtin(__builtin_add_overflow) || (defined(__GNUC__) && __GNUC__ > 5)
259         size_t result_buffer;
260         bool valid = !__builtin_add_overflow(x, y, &result_buffer);
261         if (valid && result) {
262             *result = result_buffer;
263         }
264         return valid;
265     #else
266         if (y > SIZE_MAX - x) {
267             return false;
268         }
269         if (result) {
270             *result = x + y;
271         }
272         return true;
273     #endif
274 }
275
276 /* Compute the difference of two size_t integers with overflow checking. */
277 inline static bool jas_safe_size_sub(size_t x, size_t y, size_t *result)
278 {
279     #if jas_has_builtin(__builtin_sub_overflow) || (defined(__GNUC__) && __GNUC__ > 5)
280         size_t result_buffer;
281         bool valid = !__builtin_sub_overflow(x, y, &result_buffer);
282         if (valid && result) {
283             *result = result_buffer;
284         }
285         return valid;
286     #else
287         if (y > x) {
288             return false;
289         }
290         if (result) {
291             *result = x - y;
292         }
293         return true;
294     #endif
295 }
296
297 /* Compute the product of two int_fast32_t integers with overflow checking. */
298 inline static bool jas_safe_intfast32_mul(int_fast32_t x, int_fast32_t y,
299     int_fast32_t *result)
300 {
301     #if jas_has_builtin(__builtin_mul_overflow) || (defined(__GNUC__) && __GNUC__ > 5)
302         int_fast32_t result_buffer;
303         bool valid = !__builtin_mul_overflow(x, y, &result_buffer);
304         if (valid && result) {
305             *result = result_buffer;
306         }
307         return valid;
308     #else
309         if (x > 0) {
310             /* x is positive */
311             if (y > 0) {
312                 /* x and y are positive */
313                 if (x > INT_FAST32_MAX / y) {
314                     return false;
315                 }
316             } else {
317                 /* x positive, y nonpositive */
318                 if (y < INT_FAST32_MIN / x) {
319                     return false;
320                 }
321             }
322         } else {
323             /* x is nonpositive */
324             if (y > 0) {

```

```

325             /* x is nonpositive, y is positive */
326             if (x < INT_FAST32_MIN / y) {
327                 return false;
328             }
329         } else { /* x and y are nonpositive */
330             if (x != 0 && y < INT_FAST32_MAX / x) {
331                 return false;
332             }
333         }
334     }
335
336     if (result) {
337         *result = x * y;
338     }
339     return true;
340 #endif
341 }
342
343 /* Compute the product of three int_fast32_t integers with overflow checking. */
344 inline static bool jas_safe_intfast32_mul3(int_fast32_t a, int_fast32_t b,
345     int_fast32_t c, int_fast32_t *result)
346 {
347     int_fast32_t tmp;
348     if (!jas_safe_intfast32_mul(a, b, &tmp) ||
349         !jas_safe_intfast32_mul(tmp, c, &tmp)) {
350         return false;
351     }
352     if (result) {
353         *result = tmp;
354     }
355     return true;
356 }
357
358 /* Compute the sum of two int_fast32_t integers with overflow checking. */
359 inline static bool jas_safe_intfast32_add(int_fast32_t x, int_fast32_t y,
360     int_fast32_t *result)
361 {
362     #if jas_has_builtin(__builtin_add_overflow) || (defined(__GNUC__) && __GNUC__ > 5)
363         int_fast32_t result_buffer;
364         bool valid = !__builtin_add_overflow(x, y, &result_buffer);
365         if (valid && result) {
366             *result = result_buffer;
367         }
368         return valid;
369     #else
370         if ((y > 0 && x > INT_FAST32_MAX - y) ||
371             (y < 0 && x < INT_FAST32_MIN - y)) {
372             return false;
373         }
374         if (result) {
375             *result = x + y;
376         }
377         return true;
378     #endif
379 }
380
381 #if 0
382 /*
383 This function is potentially useful but not currently used.
384 So, it is commented out.
385 */
386 inline static bool jas_safe_uint_mul(unsigned x, unsigned y, unsigned *result)
387 {
388     /* Check if overflow would occur */
389     if (x && y > UINT_MAX / x) {
390         /* Overflow would occur. */
391         return false;
392     }
393     if (result) {
394         *result = x * y;
395     }
396     return true;
397 }
398 #endif
399
400 /*****
401 * Safe integer arithmetic (i.e., with overflow checking).
402 *****/
403
404 typedef struct {
405     bool valid;

```

```

406     uint_least64_t value;
407 } jas_safeui64_t;
408
409 static inline
410 jas_safeui64_t jas_safeui64_from_intmax(intmax_t x)
411 {
412     jas_safeui64_t result;
413     if (x >= 0 && x <= UINT_LEAST64_MAX) {
414         result.valid = true;
415         result.value = JAS_CAST(uint_least64_t, x);
416     } else {
417         result.valid = false;
418         result.value = 0;
419     }
420     return result;
421 }
422
423 static inline
424 jas_safeui64_t jas_safeui64_add(jas_safeui64_t x, jas_safeui64_t y)
425 {
426     jas_safeui64_t result;
427     if (x.valid && y.valid && y.value <= UINT_LEAST64_MAX - x.value) {
428         result.valid = true;
429         result.value = x.value + y.value;
430     } else {
431         result.valid = false;
432         result.value = 0;
433     }
434     return result;
435 }
436
437 static inline
438 jas_safeui64_t jas_safeui64_sub(jas_safeui64_t x, jas_safeui64_t y)
439 {
440     jas_safeui64_t result;
441     if (x.valid && y.valid && y.value <= x.value) {
442         result.valid = true;
443         result.value = x.value - y.value;
444     } else {
445         result.valid = false;
446         result.value = 0;
447     }
448     return result;
449 }
450
451 static inline
452 jas_safeui64_t jas_safeui64_mul(jas_safeui64_t x, jas_safeui64_t y)
453 {
454     jas_safeui64_t result;
455     if (!x.valid || !y.valid || (x.value && y.value > UINT_LEAST64_MAX /
456         x.value)) {
457         result.valid = false;
458         result.value = 0;
459     } else {
460         result.valid = true;
461         result.value = x.value * y.value;
462     }
463     return result;
464 }
465
466 static inline
467 jas_safeui64_t jas_safeui64_div(jas_safeui64_t x, jas_safeui64_t y)
468 {
469     jas_safeui64_t result;
470     if (x.valid && y.valid && y.value) {
471         result.valid = true;
472         result.value = x.value / y.value;
473     } else {
474         result.valid = false;
475         result.value = 0;
476     }
477     return result;
478 }
479
480 static inline
481 jas_safeui64_t jas_safeui64_pow2_intmax(intmax_t x)
482 {
483     jas_safeui64_t result;
484     if (x >= 0 && x < 64) {
485         result.valid = true;
486         result.value = JAS_CAST(uint_least64_t, 1) << x;

```

```

487     } else {
488         result.valid = false;
489         result.value = 0;
490     }
491     return result;
492 }
493
494 static inline
495 int jas_safeui64_to_int(jas_safeui64_t x, int invalid_value)
496 {
497     int result;
498     if (x.valid && x.value <= INT_MAX) {
499         result = JAS_CAST(int, x.value);
500     } else {
501         result = invalid_value;
502     }
503     return result;
504 }
505
506 /*****
507 \*****/
508
509 #ifdef __cplusplus
510 }
511 #endif
512
513 #endif

```

16.27 jas_seq.h File Reference

Sequence/Matrix Library.

```

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

```

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.](#)

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 #ifndef JAS_SEQ_H
65 #define JAS_SEQ_H
66
67
68
69
70
71

```



```

72 /*****\
73 * Includes.
74 \*****/
75
76 /* The configuration header file should be included first. */
77 #include <jasper/jas_config.h> /* IWYU pragma: keep */
78
79 #include <jasper/jas_types.h>
80 #include <jasper/jas_math.h>
81
82 #include <stdio.h>
83
84 #ifdef __cplusplus
85 extern "C" {
86 #endif
87
88 /*****\
89 * Constants.
90 \*****/
91
92 /* This matrix is a reference to another matrix. */
93 #define JAS_MATRIX_REF 0x0001
94
95 /*****\
96 * Types.
97 \*****/
98
99 /* An element in a sequence. */
100 #ifdef JAS_ENABLE_32BIT
101 typedef int_least32_t jas_sequent_t;
102 #define PRIjas_sequent PRIiLEAST32
103 #else
104 typedef int_fast32_t jas_sequent_t;
105 #define PRIjas_sequent PRIiFAST32
106 #endif
107
108 /* An element in a matrix. */
109 #ifdef JAS_ENABLE_32BIT
110 typedef int_least32_t jas_matent_t;
111 #else
112 typedef int_fast32_t jas_matent_t;
113 #endif
114
115 #ifdef JAS_ENABLE_32BIT
116 typedef int_least32_t jas_matind_t;
117 #else
118 typedef int_fast32_t jas_matind_t;
119 #endif
120
121 typedef struct {
122     /* Additional state information. */
123     int flags_;
124
125     /* The starting horizontal index. */
126     jas_matind_t xstart_;
127
128     /* The starting vertical index. */
129     jas_matind_t ystart_;
130
131     /* The ending horizontal index. */
132     jas_matind_t xend_;
133
134     /* The ending vertical index. */
135     jas_matind_t yend_;
136
137     /* The number of rows in the matrix. */
138     jas_matind_t numRows_;
139
140     /* The number of columns in the matrix. */
141     jas_matind_t numcols_;
142
143     /* Pointers to the start of each row. */
144     jas_sequent_t **rows_;
145
146     /* The allocated size of the rows array. */
147     int_fast32_t maxrows_;
148
149     /* The matrix data buffer. */
150     jas_sequent_t *data_;
151 }

```

```

161      /* The allocated size of the data array. */
162      int_fast32_t datasize_;
163
164 } jas_matrix_t;
165
166 typedef jas_matrix_t jas_seq2d_t;
167
168 typedef jas_matrix_t jas_seq_t;
169
170 /******\
171 * Functions/macros for matrix class.
172 \*****/
173
174 JAS_ATTRIBUTE_PURE
175 static inline jas_matind_t jas_matrix_numrows(const jas_matrix_t *matrix)
176 {
177     return matrix->numrows_;
178 }
179
180 JAS_ATTRIBUTE_PURE
181 static inline jas_matind_t jas_matrix_numcols(const jas_matrix_t *matrix)
182 {
183     return matrix->numcols_;
184 }
185
186 JAS_ATTRIBUTE_PURE
187 static inline jas_matind_t jas_matrix_size(const jas_matrix_t *matrix)
188 {
189     return jas_matrix_numcols(matrix) * jas_matrix_numrows(matrix);
190 }
191
192 JAS_ATTRIBUTE_PURE
193 static inline bool jas_matrix_empty(const jas_matrix_t *matrix)
194 {
195     return jas_matrix_numcols(matrix) == 0 || jas_matrix_numrows(matrix) == 0;
196 }
197
198 JAS_ATTRIBUTE_PURE
199 static inline jas_sequent_t jas_matrix_get(const jas_matrix_t *matrix, jas_matind_t i, jas_matind_t j)
200 {
201     assert(i >= 0 && i < matrix->numrows_ && j >= 0 && j < matrix->numcols_);
202     return matrix->rows_[i][j];
203 }
204
205 static inline void jas_matrix_set(jas_matrix_t *matrix, jas_matind_t i, jas_matind_t j, jas_sequent_t v)
206 {
207     assert(i >= 0 && i < matrix->numrows_ && j >= 0 && j < matrix->numcols_);
208     matrix->rows_[i][j] = v;
209 }
210
211 JAS_ATTRIBUTE_PURE
212 static inline jas_sequent_t jas_matrix_getv(const jas_matrix_t *matrix, jas_matind_t i)
213 {
214     return matrix->numrows_ == 1
215         ? matrix->rows_[0][i]
216         : matrix->rows_[i][0];
217 }
218
219 static inline void jas_matrix_setv(jas_matrix_t *matrix, jas_matind_t i, jas_sequent_t v)
220 {
221     if (matrix->numrows_ == 1)
222         matrix->rows_[0][i] = v;
223     else
224         matrix->rows_[i][0] = v;
225 }
226
227 JAS_ATTRIBUTE_PURE
228 static inline jas_sequent_t *jas_matrix_getref(const jas_matrix_t *matrix, jas_matind_t i, jas_matind_t j)
229 {
230     return &matrix->rows_[i][j];
231 }
232
233 JAS_ATTRIBUTE_PURE
234 static inline jas_sequent_t *jas_matrix_getvref(const jas_matrix_t *matrix, jas_matind_t i)
235 {
236     return matrix->numrows_ > 1
237         ? jas_matrix_getref(matrix, i, 0)
238         : jas_matrix_getref(matrix, 0, i);
239 }
240
241 JAS_EXPORT

```

```

293 jas_matrix_t *jas_matrix_create(jas_matind_t numRows, jas_matind_t numcols);
294
299 JAS_EXPORT
300 void jas_matrix_destroy(jas_matrix_t *matrix);
301
306 JAS_EXPORT
307 int jas_matrix_resize(jas_matrix_t *matrix, jas_matind_t numRows, jas_matind_t numcols);
308
313 JAS_EXPORT
314 int jas_matrix_output(jas_matrix_t *matrix, FILE *out);
315
320 JAS_EXPORT
321 int jas_matrix_bindsub(jas_matrix_t *mat0, jas_matrix_t *mat1, jas_matind_t r0,
322   jas_matind_t c0, jas_matind_t r1, jas_matind_t c1);
323
328 static inline int jas_matrix_bindrow(jas_matrix_t *mat0, jas_matrix_t *mat1, jas_matind_t r)
329 {
330     return jas_matrix_bindsub(mat0, mat1, r, 0, r, mat1->numcols_ - 1);
331 }
332
337 static inline int jas_matrix_bindcol(jas_matrix_t *mat0, jas_matrix_t *mat1, jas_matind_t c)
338 {
339     return jas_matrix_bindsub(mat0, mat1, 0, c, mat1->numrows_ - 1, c);
340 }
341
346 JAS_EXPORT
347 void jas_matrix_clip(jas_matrix_t *matrix, jas_segent_t minval,
348   jas_segent_t maxval);
349
354 JAS_EXPORT
355 void jas_matrix_asl(jas_matrix_t *matrix, unsigned n);
356
361 JAS_EXPORT
362 void jas_matrix_asr(jas_matrix_t *matrix, unsigned n);
363
368 JAS_EXPORT
369 void jas_matrix_divpow2(jas_matrix_t *matrix, unsigned n);
370
375 JAS_EXPORT
376 void jas_matrix_setall(jas_matrix_t *matrix, jas_segent_t val);
377
382 JAS_ATTRIBUTE_PURE
383 static inline size_t jas_matrix_rowstep(const jas_matrix_t *matrix)
384 {
385     return matrix->numrows_ > 1
386         ? (size_t)(matrix->rows_[1] - matrix->rows_[0])
387         : 0u;
388 }
389
394 JAS_ATTRIBUTE_PURE
395 static inline size_t jas_matrix_step(const jas_matrix_t *matrix)
396 {
397     return matrix->numrows_ > 1
398         ? jas_matrix_rowstep(matrix)
399         : 1;
400 }
401
406 JAS_EXPORT
407 int jas_matrix_cmp(jas_matrix_t *mat0, jas_matrix_t *mat1);
408
413 JAS_EXPORT
414 jas_matrix_t *jas_matrix_copy(jas_matrix_t *x);
415
420 JAS_EXPORT
421 jas_matrix_t *jas_matrix_input(FILE *);
422
426 JAS_ATTRIBUTE_CONST
427 static inline jas_segent_t jas_segent_asl(jas_segent_t x, unsigned n)
428 {
429     #ifdef JAS_ENABLE_32BIT
430         return jas_least32_asl(x, n);
431     #else
432         return jas_fast32_asl(x, n);
433     #endif
434 }
435
439 JAS_ATTRIBUTE_CONST
440 static inline jas_segent_t jas_segent_asr(jas_segent_t x, unsigned n)
441 {
442     #ifdef JAS_ENABLE_32BIT
443         return jas_least32_asr(x, n);

```

```

444 #else
445     return jas_fast32_asr(x, n);
446 #endif
447 }
448
449 /*****
450 * Functions/macros for 2-D sequence class.
451 *****/
452
453 JAS_EXPORT
454 jas_seq2d_t *jas_seq2d_copy(jas_seq2d_t *x);
455
456 JAS_EXPORT
457 jas_matrix_t *jas_seq2d_create(jas_matind_t xstart, jas_matind_t ystart,
458     jas_matind_t xend, jas_matind_t yend);
459
460 static inline void jas_seq2d_destroy(jas_seq2d_t *s)
461 {
462     jas_matrix_destroy(s);
463 }
464
465 JAS_ATTRIBUTE_PURE
466 static inline jas_matind_t jas_seq2d_xstart(const jas_seq2d_t *s)
467 {
468     return s->xstart_;
469 }
470
471 JAS_ATTRIBUTE_PURE
472 static inline jas_matind_t jas_seq2d_ystart(const jas_seq2d_t *s)
473 {
474     return s->ystart_;
475 }
476
477 JAS_ATTRIBUTE_PURE
478 static inline jas_matind_t jas_seq2d_xend(const jas_seq2d_t *s)
479 {
480     return s->xend_;
481 }
482
483 JAS_ATTRIBUTE_PURE
484 static inline jas_matind_t jas_seq2d_yend(const jas_seq2d_t *s)
485 {
486     return s->yend_;
487 }
488
489 JAS_ATTRIBUTE_PURE
490 static inline jas_segent_t *jas_seq2d_getref(const jas_seq2d_t *s, jas_matind_t x, jas_matind_t y)
491 {
492     return jas_matrix_getref(s, y - s->ystart_, x - s->xstart_);
493 }
494
495 JAS_ATTRIBUTE_PURE
496 static inline jas_segent_t jas_seq2d_get(const jas_seq2d_t *s, jas_matind_t x, jas_matind_t y)
497 {
498     return jas_matrix_get(s, y - s->ystart_, x - s->xstart_);
499 }
500
501 JAS_ATTRIBUTE_PURE
502 static inline size_t jas_seq2d_rowstep(const jas_seq2d_t *s)
503 {
504     return jas_matrix_rowstep(s);
505 }
506
507 JAS_ATTRIBUTE_PURE
508 static inline unsigned jas_seq2d_width(const jas_seq2d_t *s)
509 {
510     return (unsigned)(s->xend_ - s->xstart_);
511 }
512
513 JAS_ATTRIBUTE_PURE
514 static inline unsigned jas_seq2d_height(const jas_seq2d_t *s)
515 {
516     return (unsigned)(s->yend_ - s->ystart_);
517 }
518
519 static inline void jas_seq2d_setshift(jas_seq2d_t *s, jas_matind_t x, jas_matind_t y)
520 {
521     s->xstart_ = x;
522     s->ystart_ = y;
523     s->xend_ = s->xstart_ + s->numcols_;
524     s->yend_ = s->ystart_ + s->numrows_;
525 }

```

```

577 }
578
583 JAS_ATTRIBUTE_PURE
584 static inline jas_matind_t jas_seq2d_size(const jas_seq2d_t *s)
585 {
586     return jas_seq2d_width(s) * jas_seq2d_height(s);
587 }
588
593 JAS_ATTRIBUTE_PURE
594 static inline bool jas_seq2d_empty(const jas_seq2d_t *s)
595 {
596     return jas_seq2d_width(s) == 0 || jas_seq2d_height(s) == 0;
597 }
598
603 JAS_EXPORT
604 int jas_seq2d_bindsub(jas_matrix_t *s, jas_matrix_t *s1, jas_matind_t xstart,
605     jas_matind_t ystart, jas_matind_t xend, jas_matind_t yend);
606
607 /*****
608  * Functions/macros for 1-D sequence class.
609  *****/
610
615 static inline jas_seq_t *jas_seq_create(jas_matind_t start, jas_matind_t end)
616 {
617     return jas_seq2d_create(start, 0, end, 1);
618 }
619
624 static inline void jas_seq_destroy(jas_seq_t *seq)
625 {
626     jas_seq2d_destroy(seq);
627 }
628
633 static inline void jas_seq_set(jas_seq_t *seq, jas_matind_t i, jas_seqent_t v)
634 {
635     seq->rows_[0][i - seq->xstart_] = v;
636 }
637
642 JAS_ATTRIBUTE_PURE
643 static inline jas_seqent_t *jas_seq_getref(const jas_seq_t *seq, jas_matind_t i)
644 {
645     return &seq->rows_[0][i - seq->xstart_];
646 }
647
652 JAS_ATTRIBUTE_PURE
653 static inline jas_seqent_t jas_seq_get(const jas_seq_t *seq, jas_matind_t i)
654 {
655     return seq->rows_[0][i - seq->xstart_];
656 }
657
662 JAS_ATTRIBUTE_PURE
663 static inline jas_matind_t jas_seq_start(const jas_seq_t *seq)
664 {
665     return seq->xstart_;
666 }
667
672 JAS_ATTRIBUTE_PURE
673 static inline jas_matind_t jas_seq_end(const jas_seq_t *seq)
674 {
675     return seq->xend_;
676 }
677
682 #ifdef __cplusplus
683 }
684 #endif
685
686 #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) (((stream)->flags_ & JAS_STREAM_EOF) != 0)
Get the EOF indicator for a stream.
- #define [jas_stream_error](#)(stream) (((stream)->flags_ & JAS_STREAM_ERR) != 0)
Get the error indicator for a stream.
- #define [jas_stream_clearerr](#)(stream) ((stream)->flags_ &= ~(JAS_STREAM_ERR | JAS_STREAM_EOF))
Clear the error indicator for a stream.
- #define [jas_stream_getrwlmit](#)(stream) (((const [jas_stream_t](#) *) (stream))->rwlmit_)
Get the read/write limit for a stream.
- #define [jas_stream_getrwcoun](#)(stream) (((const [jas_stream_t](#) *) (stream))->rwcoun_)
Get the read/write count for a stream.
- #define [jas_stream_getc](#)(stream) [jas_stream_getc_func](#)(stream)
jas_stream_getc Read a character from a stream.
- #define [jas_stream_putc](#)(stream, c) [jas_stream_putc_func](#)(stream, c)
jas_stream_putc Write a character to a stream.
- #define [jas_stream_peekc](#)(stream)
Look at the next character to be read from a stream without actually removing the character from the stream.

Functions

- JAS_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_coun)
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.](#)

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2003 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 #ifndef JAS_STREAM_H
65 #define JAS_STREAM_H
66
67 /*****\
68 * Includes.
69 \*****/
70
71 /* The configuration header file should be included first. */
72 #include <jasper/jas_config.h> /* IWYU pragma: export */
73
74 #include <stdio.h>
75 #if defined(JAS_HAVE_FCNTL_H)
76 #include <fcntl.h>

```



```

82 #endif
83 #include <jasper/jas_types.h>
84
85 #ifdef __cplusplus
86 extern "C" {
87 #endif
88
89 /*****
90  * Constants.
91  *****/
92
93 /* On most UNIX systems, we probably need to define O_BINARY ourselves. */
94 #ifndef O_BINARY
95 #define O_BINARY 0
96 #endif
97
98 /*
99  * Stream open flags.
100  */
101
102 /* The stream was opened for reading. */
103 #define JAS_STREAM_READ 0x0001
104 /* The stream was opened for writing. */
105 #define JAS_STREAM_WRITE 0x0002
106 /* The stream was opened for appending. */
107 #define JAS_STREAM_APPEND 0x0004
108 /* The stream was opened in binary mode. */
109 #define JAS_STREAM_BINARY 0x0008
110 /* The stream should be created/truncated. */
111 #define JAS_STREAM_CREATE 0x0010
112
113 /*
114  * Stream buffering flags.
115  */
116
117 /* The stream is unbuffered. */
118 #define JAS_STREAM_UNBUF 0x0000
119 /* The stream is line buffered. */
120 #define JAS_STREAM_LINEBUF 0x0001
121 /* The stream is fully buffered. */
122 #define JAS_STREAM_FULLBUF 0x0002
123 /* The buffering mode mask. */
124 #define JAS_STREAM_BUFMODEMASK 0x000f
125
126 /* The memory associated with the buffer needs to be deallocated when the
127  * stream is destroyed. */
128 #define JAS_STREAM_FREEBUF 0x0008
129 /* The buffer is currently being used for reading. */
130 #define JAS_STREAM_RDBUF 0x0010
131 /* The buffer is currently being used for writing. */
132 #define JAS_STREAM_WRBUF 0x0020
133
134 /*
135  * Stream error flags.
136  */
137
138 /* The end-of-file has been encountered (on reading). */
139 #define JAS_STREAM_EOF 0x0001
140 /* An I/O error has been encountered on the stream. */
141 #define JAS_STREAM_ERR 0x0002
142 /* The read/write limit has been exceeded. */
143 #define JAS_STREAM_RWLIMIT 0x0004
144 /* The error mask. */
145 #define JAS_STREAM_ERRMASK \
146     (JAS_STREAM_EOF | JAS_STREAM_ERR | JAS_STREAM_RWLIMIT)
147
148 /*
149  * Other miscellaneous constants.
150  */
151
152 /* The default buffer size (for fully-buffered operation). */
153 #define JAS_STREAM_BUFSIZE 8192
154 /* The default permission mask for file creation. */
155 #define JAS_STREAM_PERMS 0666
156
157 /* The maximum number of characters that can always be put back on a stream. */
158 #define JAS_STREAM_MAXPUTBACK 16
159
160 /*****
161  * Types.
162  *****/

```

```

168
169 /*
170  * Generic file object.
171  */
172
173 typedef void jas_stream_obj_t;
174
175 /*
176  * Generic file object operations.
177  */
178
179 typedef struct {
180
181     /* Read characters from a file object. */
182     ssize_t (*read_)(jas_stream_obj_t *obj, char *buf, size_t cnt);
183
184     /* Write characters to a file object. */
185     ssize_t (*write_)(jas_stream_obj_t *obj, const char *buf, size_t cnt);
186
187     /* Set the position for a file object. */
188     long (*seek_)(jas_stream_obj_t *obj, long offset, int origin);
189
190     /* Close a file object. */
191     int (*close_)(jas_stream_obj_t *obj);
192 } jas_stream_ops_t;
193
194
195 typedef struct {
196
197     /* The mode in which the stream was opened. */
198     int openmode_;
199
200     /* The buffering mode. */
201     int bufmode_;
202
203     /* The stream status. */
204     int flags_;
205
206     /* The start of the buffer area to use for reading/writing. */
207     jas_uchar *bufbase_;
208
209     /* The start of the buffer area excluding the extra initial space for
210        character putback. */
211     jas_uchar *bufstart_;
212
213     /* The buffer size. */
214     int bufsize_;
215
216     /* The current position in the buffer. */
217     jas_uchar *ptr_;
218
219     /* The number of characters that must be read/written before
220        the buffer needs to be filled/flushed. */
221     int cnt_;
222
223     /* A trivial buffer to be used for unbuffered operation. */
224     jas_uchar tinybuf_[JAS_STREAM_MAXPUTBACK + 1];
225
226     /* The operations for the underlying stream file object. */
227     const jas_stream_ops_t *ops_;
228
229     /* The underlying stream file object. */
230     jas_stream_obj_t *obj_;
231
232     /* The number of characters read/written. */
233     long rwcnt_;
234
235     /* The maximum number of characters that may be read/written. */
236     long rwlimit_;
237 } jas_stream_t;
238
239 /*
240  * Regular file object.
241  */
242
243 /*
244  * File descriptor file object.
245  */
246
247 typedef struct {
248     int fd;

```

```

260         int flags;
261         char pathname[L_tmpnam + 1];
262 } jas_stream_fileobj_t;
263
264 /* Delete underlying file object upon stream close. */
265 #define JAS_STREAM_FILEOBJ_DELCLOSE 0x01
266 /* Do not close underlying file object upon stream close. */
267 #define JAS_STREAM_FILEOBJ_NOCLOSE 0x02
268
269 /*
270  * Memory file object.
271  */
272
273 typedef struct {
274
275     /* The data associated with this file. */
276     jas_uchar *buf_;
277
278     /* The allocated size of the buffer for holding file data. */
279     size_t bufsize_;
280
281     /* The length of the file. */
282     size_t len_;
283
284     /* The seek position. */
285     size_t pos_;
286
287     /* Is the buffer growable? */
288     int growable_;
289
290     /* Was the buffer allocated internally? */
291     int myalloc_;
292
293 } jas_stream_memobj_t;
294
295 /*****
296  * Macros/functions for opening and closing streams.
297  *****/
298
313 JAS_EXPORT
314 jas_stream_t *jas_stream_fopen(const char *filename, const char *mode);
315
340 JAS_EXPORT
341 jas_stream_t *jas_stream_memopen(char *buffer, size_t buffer_size);
342
351 JAS_DEPRECATED
352 JAS_EXPORT
353 jas_stream_t *jas_stream_memopen2(char *buffer, size_t buffer_size);
354
369 JAS_EXPORT
370 jas_stream_t *jas_stream_fdopen(int fd, const char *mode);
371
394 JAS_EXPORT
395 jas_stream_t *jas_stream_freopen(const char *path, const char *mode, FILE *fp);
396
411 JAS_EXPORT
412 jas_stream_t *jas_stream_tmpfile(void);
413
432 JAS_EXPORT
433 int jas_stream_close(jas_stream_t *stream);
434
435 /*****
436  * Macros/functions for getting/setting the stream state.
437  *****/
438
449 #define jas_stream_eof(stream) \
450     (((stream)->flags_ & JAS_STREAM_EOF) != 0)
451
463 #define jas_stream_error(stream) \
464     (((stream)->flags_ & JAS_STREAM_ERR) != 0)
465
475 #define jas_stream_clearerr(stream) \
476     ((stream)->flags_ &= ~(JAS_STREAM_ERR | JAS_STREAM_EOF))
477
488 #define jas_stream_getrwlimit(stream) \
489     (((const jas_stream_t *) (stream))->rwlimit_)
490
506 JAS_EXPORT long jas_stream_setrwlimit(jas_stream_t *stream, long rwlimit);
507
518 #define jas_stream_getrwcoun(stream) \
519     (((const jas_stream_t *) (stream))->rwcnt_)

```

```

520
534 JAS_EXPORT
535 long jas_stream_setrwcoun(jas_stream_t *stream, long rw_count);
536
537 /*****
538 * Macros/functions for I/O.
539 *****/
540
541 /* Read a character from a stream. */
542 #ifndef NDEBUG
543 #define jas_stream_getc(stream) jas_stream_getc_func(stream)
544 #else
545 #define jas_stream_getc(stream) jas_stream_getc_macro(stream)
546 #endif
547
548
549 /* Write a character to a stream. */
550 #ifndef NDEBUG
551 #define jas_stream_putc(stream, c) jas_stream_putc_func(stream, c)
552 #else
553 #define jas_stream_putc(stream, c) jas_stream_putc_macro(stream, c)
554 #endif
555
556 JAS_EXPORT
557 size_t jas_stream_read(jas_stream_t *stream, void *buffer, size_t count);
558
559 JAS_EXPORT
560 unsigned jas_stream_peek(jas_stream_t *stream, void *buffer, size_t count);
561
562 JAS_EXPORT
563 size_t jas_stream_write(jas_stream_t *stream, const void *buffer,
564     size_t count);
565
566 JAS_EXPORT
567 int jas_stream_printf(jas_stream_t *stream, const char *format, ...);
568
569 JAS_EXPORT
570 int jas_stream_puts(jas_stream_t *stream, const char *s);
571
572 JAS_EXPORT
573 char *jas_stream_gets(jas_stream_t *stream, char *buffer, int buffer_size);
574
575 #define jas_stream_peekc(stream) \
576     (((stream)->cnt_ <= 0) ? jas_stream_fillbuf(stream, 0) : \
577     ((int) (*(stream)->ptr_)))
578
579 JAS_EXPORT
580 int jas_stream_ungetc(jas_stream_t *stream, int c);
581
582 /*****
583 * Macros/functions for getting/setting the stream position.
584 *****/
585
586 JAS_EXPORT
587 JAS_ATTRIBUTE_PURE
588 int jas_stream_isseekable(jas_stream_t *stream);
589
590 JAS_EXPORT
591 long jas_stream_seek(jas_stream_t *stream, long offset, int origin);
592
593 JAS_EXPORT
594 long jas_stream_tell(jas_stream_t *stream);
595
596 JAS_EXPORT
597 int jas_stream_rewind(jas_stream_t *stream);
598
599 /*****
600 * Macros/functions for flushing.
601 *****/
602
603 JAS_EXPORT
604 int jas_stream_flush(jas_stream_t *stream);
605
606 /*****
607 * Miscellaneous macros/functions.
608 *****/
609
610 JAS_EXPORT
611 int jas_stream_copy(jas_stream_t *destination, jas_stream_t *source,
612     ssize_t count);
613
614 JAS_EXPORT

```

```

965 int jas_stream_display(jas_stream_t *stream, FILE *fp, int count);
966
967 JAS_EXPORT
968 ssize_t jas_stream_gobble(jas_stream_t *stream, size_t count);
969
970 JAS_EXPORT
971 ssize_t jas_stream_pad(jas_stream_t *stream, size_t count, int value);
972
973 JAS_EXPORT
974 long jas_stream_length(jas_stream_t *stream);
975
976 /*****
977  * Internal functions.
978  *****/
979
980 /* The following functions are for internal use only! If you call them
981 directly, you will die a horrible, miserable, and painful death! */
982
983 /* These prototypes need to be here for the sake of the stream_getc and
984 stream_putc macros. */
985
986 /* Library users must not invoke these functions directly. */
987 JAS_EXPORT int jas_stream_fillbuf(jas_stream_t *stream, int getflag);
988 JAS_EXPORT int jas_stream_flushbuf(jas_stream_t *stream, int c);
989 JAS_EXPORT int jas_stream_getc_func(jas_stream_t *stream);
990 JAS_EXPORT int jas_stream_putc_func(jas_stream_t *stream, int c);
991
992 /* Read a character from a stream. */
993 static inline int jas_stream_getc2(jas_stream_t *stream)
994 {
995     if (--stream->cnt_ < 0)
996         return jas_stream_fillbuf(stream, 1);
997
998     ++stream->rwcnt_;
999     return (int) (*stream->ptr_++);
1000 }
1001
1002 static inline int jas_stream_getc_macro(jas_stream_t *stream)
1003 {
1004     if (stream->flags_ & (JAS_STREAM_ERR | JAS_STREAM_EOF | JAS_STREAM_RWLIMIT))
1005         return EOF;
1006
1007     if (stream->rwlimit_ >= 0 && stream->rwcnt_ >= stream->rwlimit_) {
1008         stream->flags_ |= JAS_STREAM_RWLIMIT;
1009         return EOF;
1010     }
1011
1012     return jas_stream_getc2(stream);
1013 }
1014
1015 /* Write a character to a stream. */
1016 static inline int jas_stream_putc2(jas_stream_t *stream, jas_uchar c)
1017 {
1018     stream->bufmode_ |= JAS_STREAM_WRBUFF;
1019
1020     if (--stream->cnt_ < 0)
1021         return jas_stream_flushbuf(stream, c);
1022     else {
1023         ++stream->rwcnt_;
1024         return (int) (*stream->ptr_++ = c);
1025     }
1026 }
1027
1028 static inline int jas_stream_putc_macro(jas_stream_t *stream, jas_uchar c)
1029 {
1030     if (stream->flags_ & (JAS_STREAM_ERR | JAS_STREAM_EOF | JAS_STREAM_RWLIMIT))
1031         return EOF;
1032
1033     if (stream->rwlimit_ >= 0 && stream->rwcnt_ >= stream->rwlimit_) {
1034         stream->flags_ |= JAS_STREAM_RWLIMIT;
1035         return EOF;
1036     }
1037
1038     return jas_stream_putc2(stream, c);
1039 }
1040
1041 #ifdef __cplusplus
1042 }
1043 #endif
1044 #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.](#)

```
1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
```

```

33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 #ifndef JAS_STRING_H
65 #define JAS_STRING_H
66
67 /*****
68  * Includes.
69  *****/
70 /* The configuration header file should be included first. */
71 #include <jasper/jas_config.h>
72
73 #include <jasper/jas_types.h>
74
75 #ifdef __cplusplus
76 extern "C" {
77 #endif
78
79 /*****
80  * Functions.
81  *****/
82
83 JAS_EXPORT
84 char *jas_strdup(const char *);
85
86 JAS_EXPORT
87 char *jas_strtok(char *str, const char *delim, char **saveptr);
88
89 JAS_EXPORT
90 int jas_stringtokenize(const char *string, const char *delim,
91 char ***tokens_buf, size_t *max_tokens_buf, size_t *num_tokens_buf);
92
93 #ifdef __cplusplus
94 }
95 #endif
96 #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.](#)

```
1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2003 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
```



```

54  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60  *
61  * __END_OF_JASPER_LICENSE__
62  */
63
69 #ifndef JAS_THREAD_H
70 #define JAS_THREAD_H
71
72 /*****\
73  * Includes
74  \*****/
75
76 /* The configuration header file should be included first. */
77 #include <jasper/jas_config.h>
78
79 #include "jasper/jas_compiler.h"
80 #include "jasper/jas_types.h"
81
82 #if defined(JAS_THREADS)
83
84 #include <stdlib.h>
85 #include <assert.h>
86
87 #if defined(JAS_THREADS_C11)
88 #include <threads.h>
89 #include <stdatomic.h>
90 #elif defined(JAS_THREADS_PTHREAD)
91 #include <pthread.h>
92 #include <sched.h>
93 #elif defined(JAS_THREADS_WIN32)
94 #include <process.h>
95 #include <windows.h>
96 #include <processthreadsapi.h>
97 #endif
98
99 #endif
100
101 /*****\
102  \*****/
103
104 #ifdef __cplusplus
105 extern "C" {
106 #endif
107
108 #if defined(JAS_THREADS)
109 #if defined(JAS_FOR_INTERNAL_USE_ONLY) || defined(JAS_FOR_JASPER_APP_USE_ONLY)
110
111 /*****\
112  * Types
113  \*****/
114
115 #if defined(JAS_THREADS_C11)
116 #define JAS_THREADS_IMPL "C11"
117 #define JAS_USE_SPINLOCK
118 #elif defined(JAS_THREADS_PTHREAD)
119 #define JAS_THREADS_IMPL "PTHREAD"
120 #undef JAS_USE_SPINLOCK
121 #elif defined(JAS_THREADS_WIN32)
122 #define JAS_THREADS_IMPL "WIN32"
123 #define JAS_USE_SPINLOCK
124 #endif
125
126 /*****\
127  * Spinlock
128  \*****/
129
130 #if defined(JAS_THREADS_C11)
131 #define JAS_USE_SPINLOCK
132 typedef struct {
133     atomic_flag flag;
134 } jas_spinlock_t;
135 #elif defined(JAS_THREADS_PTHREAD)
136 /* There is no pthread_spinlock_t type on MacOS. */
137 #undef JAS_USE_SPINLOCK
138 #elif defined(JAS_THREADS_WIN32)
139 #define JAS_USE_SPINLOCK
140

```

```

147 typedef struct {
148     LONG flag;
149 } jas_spinlock_t;
150 #endif
151
152 #if defined(JAS_THREADS_C11)
153 #define JAS_SPINLOCK_INITIALIZER {ATOMIC_FLAG_INIT}
154 #elif defined(JAS_THREADS_PTHREAD)
155 /* There is no pthread_spinlock_t type on MacOS. */
156 #elif defined(JAS_THREADS_WIN32)
157 #define JAS_SPINLOCK_INITIALIZER {0}
158 #endif
159
160 /*****
161  * Basic Mutex
162  *****/
163
164 #if defined(JAS_THREADS_C11)
165 typedef mtx_t jas_basicmutex_t;
166 #elif defined(JAS_THREADS_PTHREAD)
167 typedef pthread_mutex_t jas_basicmutex_t;
168 #elif defined(JAS_THREADS_WIN32)
169 typedef CRITICAL_SECTION jas_basicmutex_t;
170 #endif
171
172 #if defined(JAS_THREADS_C11)
173 #undef JAS_BASICMUTEX_INITIALIZER
174 #elif defined(JAS_THREADS_PTHREAD)
175 #define JAS_BASICMUTEX_INITIALIZER PTHREAD_MUTEX_INITIALIZER
176 #elif defined(JAS_THREADS_WIN32)
177 #define JAS_BASICMUTEX_INITIALIZER
178 #endif
179
180 /*****
181  * Mutex (Allowing Static Initialization)
182  *****/
183
184 #if defined(JAS_USE_SPINLOCK)
185 #define jas_mutex_t jas_spinlock_t
186 #define JAS_MUTEX_INITIALIZER JAS_SPINLOCK_INITIALIZER
187 #define jas_mutex_init jas_spinlock_init
188 #define jas_mutex_cleanup jas_spinlock_cleanup
189 #define jas_mutex_lock jas_spinlock_lock
190 #define jas_mutex_unlock jas_spinlock_unlock
191 #else
192 #define jas_mutex_t jas_basicmutex_t
193 #define JAS_MUTEX_INITIALIZER JAS_BASICMUTEX_INITIALIZER
194 #define jas_mutex_init jas_basicmutex_init
195 #define jas_mutex_cleanup jas_basicmutex_cleanup
196 #define jas_mutex_lock jas_basicmutex_lock
197 #define jas_mutex_unlock jas_basicmutex_unlock
198 #endif
199
200 /*****
201  * Once Flag
202  *****/
203
204 #if defined(JAS_THREADS_C11)
205 typedef once_flag jas_once_flag_t;
206 #elif defined(JAS_THREADS_PTHREAD)
207 typedef pthread_once_t jas_once_flag_t;
208 #elif defined(JAS_THREADS_WIN32)
209 typedef struct {
210     volatile LONG status;
211 } jas_once_flag_t;
212 #endif
213
214 #if defined(JAS_THREADS_C11)
215 #define JAS_ONCE_FLAG_INIT ONCE_FLAG_INIT
216 #elif defined(JAS_THREADS_PTHREAD)
217 #define JAS_ONCE_FLAG_INIT PTHREAD_ONCE_INIT
218 #elif defined(JAS_THREADS_WIN32)
219 #define JAS_ONCE_FLAG_INIT {0}
220 #endif
221
222 /*****
223  * Threads
224  *****/
225
226 #if defined(JAS_FOR_INTERNAL_USE_ONLY) || defined(JAS_FOR_JASPER_APP_USE_ONLY)
227

```

```

234 #if defined(JAS_THREADS_C11)
235 typedef thrd_t jas_thread_id_t;
236 #elif defined(JAS_THREADS_PTHREAD)
237 typedef pthread_t jas_thread_id_t;
238 #elif defined(JAS_THREADS_WIN32)
239 typedef HANDLE jas_thread_id_t;
240 #endif
241
242 #if defined(JAS_THREADS_C11)
243 typedef thrd_t jas_thread_t;
244 #elif defined(JAS_THREADS_PTHREAD)
245 typedef struct {
246     pthread_t id;
247     int (*func)(void *);
248     void *arg;
249     int result;
250 } jas_thread_t;
251 #elif defined(JAS_THREADS_WIN32)
252 typedef struct {
253     jas_thread_id_t id;
254     int (*func)(void *);
255     void *arg;
256 } jas_thread_t;
257 #endif
258
259 static inline void jas_thread_yield(void);
260
261 #endif
262
263 /*****\
264  * Thread-Specific Storage (TSS)
265  \*****/
266
267 #if defined(JAS_THREADS_C11)
268 typedef tss_t jas_tss_t;
269 #elif defined(JAS_THREADS_PTHREAD)
270 typedef pthread_key_t jas_tss_t;
271 #elif defined(JAS_THREADS_WIN32)
272 typedef DWORD jas_tss_t;
273 #endif
274
275 /*****\
276  * Spinlock
277  \*****/
278
279 #if defined(JAS_USE_SPINLOCK)
280
281 static inline int jas_spinlock_init(jas_spinlock_t *mtx)
282 {
283     assert(mtx);
284     #if defined(JAS_THREADS_C11)
285         atomic_flag_clear(&mtx->flag);
286         return 0;
287     #elif defined(JAS_THREADS_PTHREAD)
288         JAS_UNUSED(mtx);
289         abort();
290         return -1;
291     #elif defined(JAS_THREADS_WIN32)
292         InterlockedExchange(&mtx->flag, 0);
293         return 0;
294     #endif
295 }
296
297 static inline int jas_spinlock_cleanup(jas_spinlock_t *mtx)
298 {
299     assert(mtx);
300     #if defined(JAS_THREADS_C11)
301         JAS_UNUSED(mtx);
302         return 0;
303     #elif defined(JAS_THREADS_PTHREAD)
304         JAS_UNUSED(mtx);
305         abort();
306         return -1;
307     #elif defined(JAS_THREADS_WIN32)
308         JAS_UNUSED(mtx);
309         return 0;
310     #endif
311 }
312
313 static inline int jas_spinlock_lock(jas_spinlock_t *mtx)

```

```

350 {
351     assert(mtx);
352 #if defined(JAS_THREADS_C11)
353     while (atomic_flag_test_and_set(&mtx->flag)) {}
354     return 0;
355 #elif defined(JAS_THREADS_PTHREAD)
356     JAS_UNUSED(mtx);
357     abort();
358     return -1;
359 #elif defined(JAS_THREADS_WIN32)
360     while (InterlockedCompareExchange(&mtx->flag, 1, 0)) {}
361     return 0;
362 #endif
363 }
364
365 static inline int jas_spinlock_unlock(jas_spinlock_t *mtx)
366 {
367     assert(mtx);
368 #if defined(JAS_THREADS_C11)
369     atomic_flag_clear(&mtx->flag);
370     return 0;
371 #elif defined(JAS_THREADS_PTHREAD)
372     JAS_UNUSED(mtx);
373     abort();
374     return -1;
375 #elif defined(JAS_THREADS_WIN32)
376     InterlockedExchange(&mtx->flag, 0);
377     return 0;
378 #endif
379 }
380
381 #endif
382
383 /*****
384  * Basic Mutex
385  *****/
386
387 /* For internal use only. */
388 static inline int jas_basicmutex_init(jas_basicmutex_t *mtx)
389 {
390     assert(mtx);
391 #if defined(JAS_THREADS_C11)
392     return mtx_init(mtx, mtx_plain) == thrd_success ? 0 : -1;
393 #elif defined(JAS_THREADS_PTHREAD)
394     return pthread_mutex_init(mtx, 0);
395 #elif defined(JAS_THREADS_WIN32)
396     InitializeCriticalSection(mtx);
397     return 0;
398 #endif
399 }
400
401 /* For internal use only. */
402 static inline int jas_basicmutex_cleanup(jas_basicmutex_t *mtx)
403 {
404     assert(mtx);
405 #if defined(JAS_THREADS_C11)
406     mtx_destroy(mtx);
407     return 0;
408 #elif defined(JAS_THREADS_PTHREAD)
409     return pthread_mutex_destroy(mtx);
410 #elif defined(JAS_THREADS_WIN32)
411     DeleteCriticalSection(mtx);
412     return 0;
413 #endif
414 }
415
416 /* For internal use only. */
417 static inline int jas_basicmutex_lock(jas_basicmutex_t *mtx)
418 {
419     assert(mtx);
420 #if defined(JAS_THREADS_C11)
421     return mtx_lock(mtx);
422 #elif defined(JAS_THREADS_PTHREAD)
423     return pthread_mutex_lock(mtx);
424 #elif defined(JAS_THREADS_WIN32)
425     EnterCriticalSection(mtx);
426     return 0;
427 #endif
428 }
429
430 /* For internal use only. */

```

```

442 static inline int jas_basicmutex_unlock(jas_basicmutex_t *mtx)
443 {
444     assert(mtx);
445 #if defined(JAS_THREADS_C11)
446     return mtx_unlock(mtx);
447 #elif defined(JAS_THREADS_PTHREAD)
448     return pthread_mutex_unlock(mtx);
449 #elif defined(JAS_THREADS_WIN32)
450     LeaveCriticalSection(mtx);
451     return 0;
452 #endif
453 }
454
455 /*****
456  * Thread-Specific Storage (TSS)
457  *****/
458
459 static inline
460 int jas_tss_create(jas_tss_t *tss, void (*destructor)(void *))
461 {
462     assert(tss);
463 #if defined(JAS_THREADS_C11)
464     return tss_create(tss, destructor) == thrd_success ? 0 : -1;
465 #elif defined(JAS_THREADS_PTHREAD)
466     return pthread_key_create(tss, destructor);
467 #elif defined(JAS_THREADS_WIN32)
468     if (destructor) {
469         return -1;
470     }
471     DWORD id;
472     if ((id = TlsAlloc()) == TLS_OUT_OF_INDEXES) {
473         return -2;
474     }
475     *tss = id;
476     return 0;
477 #endif
478 }
479
480 static inline
481 void jas_tss_delete(jas_tss_t tss)
482 {
483 #if defined(JAS_THREADS_C11)
484     tss_delete(tss);
485 #elif defined(JAS_THREADS_PTHREAD)
486     pthread_key_delete(tss);
487 #elif defined(JAS_THREADS_WIN32)
488     TlsFree(tss);
489 #endif
490 }
491
492 static inline
493 void *jas_tss_get(jas_tss_t tss)
494 {
495 #if defined(JAS_THREADS_C11)
496     return tss_get(tss);
497 #elif defined(JAS_THREADS_PTHREAD)
498     return pthread_getspecific(tss);
499 #elif defined(JAS_THREADS_WIN32)
500     return TlsGetValue(tss);
501 #endif
502 }
503
504 static inline
505 int jas_tss_set(jas_tss_t tss, void *value)
506 {
507 #if defined(JAS_THREADS_C11)
508     return tss_set(tss, value) == thrd_success ? 0 : -1;
509 #elif defined(JAS_THREADS_PTHREAD)
510     return pthread_setspecific(tss, value);
511 #elif defined(JAS_THREADS_WIN32)
512     return TlsSetValue(tss, value) ? 0 : -1;
513 #endif
514 }
515
516 /*****
517  * Once Flag
518  *****/
519
520 static inline int jas_call_once(jas_once_flag_t *flag, void (*func)(void))
521 {
522     assert(flag);

```

```

575     assert(func);
576 #if defined(JAS_THREADS_C11)
577     call_once(flag, func);
578     return 0;
579 #elif defined(JAS_THREADS_PTHREAD)
580     return pthread_once(flag, func);
581 #elif defined(JAS_THREADS_WIN32)
582     if (InterlockedCompareExchange(&flag->status, 1, 0) == 0) {
583         (func)();
584         InterlockedExchange(&flag->status, 2);
585     } else {
586         while (flag->status == 1) {
587             /* Perform a busy wait. This is ugly. */
588             /* Yield processor. */
589             SwitchToThread();
590         }
591     }
592     return 0;
593 #endif
594 }
595
596 /*****
597 * Threads
598 *****/
599
600 #if defined(JAS_FOR_INTERNAL_USE_ONLY) || defined(JAS_FOR_JASPER_APP_USE_ONLY)
601
602 #if defined(JAS_THREADS_PTHREAD)
603 static void *thread_func_wrapper(void *thread_ptr)
604 {
605     jas_thread_t *thread = JAS_CAST(jas_thread_t *, thread_ptr);
606     int result = (thread->func)(thread->arg);
607     thread->result = result;
608     return thread;
609 }
610 #elif defined(JAS_THREADS_WIN32)
611 static unsigned __stdcall thread_func_wrapper(void *thread_ptr)
612 {
613     jas_thread_t *thread = JAS_CAST(jas_thread_t *, thread_ptr);
614     int result = (thread->func)(thread->arg);
615     return JAS_CAST(unsigned, result);
616 }
617 #endif
618
619 static inline
620 int jas_thread_compare(jas_thread_id_t x, jas_thread_id_t y)
621 {
622 #if defined(JAS_THREADS_C11)
623     return thrd_equal(x, y);
624 #elif defined(JAS_THREADS_PTHREAD)
625     return pthread_equal(x, y);
626 #elif defined(JAS_THREADS_WIN32)
627     return GetThreadId(x) == GetThreadId(y);
628 #endif
629 }
630
631 static inline
632 int jas_thread_create(jas_thread_t *thread, int (*func)(void *), void *arg)
633 {
634     assert(thread);
635     assert(func);
636 #if defined(JAS_THREADS_C11)
637     return thrd_create(thread, func, arg) == thrd_success ? 0 : -1;
638 #elif defined(JAS_THREADS_PTHREAD)
639     thread->func = func;
640     thread->arg = arg;
641     thread->result = 0;
642     return pthread_create(&thread->id, 0, thread_func_wrapper, thread);
643 #elif defined(JAS_THREADS_WIN32)
644     uintptr_t handle;
645     thread->func = func;
646     thread->arg = arg;
647     if (! (handle = _beginthreadex(0, 0, thread_func_wrapper, thread, 0, 0))) {
648         return -1;
649     }
650     thread->id = JAS_CAST(jas_thread_id_t, handle);
651     return 0;
652 #endif
653 }
654
655 static inline

```

```

680 int jas_thread_join(jas_thread_t *thread, int *result)
681 {
682     assert(thread);
683 #if defined(JAS_THREADS_C11)
684     return thrd_join(*thread, result) == thrd_success ? 0 : -1;
685 #elif defined(JAS_THREADS_PTHREAD)
686     void *result_buf;
687     int ret = pthread_join(thread->id, &result_buf);
688     if (!ret) {
689         jas_thread_t *other_thread = JAS_CAST(jas_thread_t *, result_buf);
690         if (result) {
691             /* A null pointer is probably a bug. */
692             assert(other_thread);
693             *result = other_thread ? other_thread->result : 0;
694         }
695     }
696     return ret;
697 #elif defined(JAS_THREADS_WIN32)
698     DWORD w;
699     DWORD code;
700     if ((w = WaitForSingleObject(thread->id, INFINITE)) != WAIT_OBJECT_0) {
701         return -1;
702     }
703     if (result) {
704         if (!GetExitCodeThread(thread->id, &code)) {
705             CloseHandle(thread->id);
706             return -1;
707         }
708         *result = JAS_CAST(int, code);
709     }
710     CloseHandle(thread->id);
711     return 0;
712 #endif
713 }
714
725 static inline void jas_thread_yield(void)
726 {
727 #if defined(JAS_THREADS_C11)
728     thrd_yield();
729 #elif defined(JAS_THREADS_PTHREAD)
730     sched_yield();
731 #elif defined(JAS_THREADS_WIN32)
732     SwitchToThread();
733 #endif
734 }
735
736 #if 0
737 /* This functionality is not available for all threading support libraries. */
738 static inline
739 void jas_thread_exit(int result)
740 {
741 #if defined(JAS_THREADS_C11)
742     thrd_exit(result);
743 #elif defined(JAS_THREADS_PTHREAD)
744     /* This does not have a trivial implementation, as far as I can see. */
745     /* There is no jas_thread_find function. */
746     jas_thread_t *thread = jas_thread_find(pthread_self());
747     thread->result = result;
748     pthread_exit(JAS_CAST(void *, thread));
749 #endif
750 }
751 #endif
752
753 #if 0
754 static inline
755 jas_thread_id_t jas_thread_current(void)
756 {
757 #if defined(JAS_THREADS_C11)
758     return thrd_current();
759 #elif defined(JAS_THREADS_PTHREAD)
760     return pthread_self();
761 #elif defined(JAS_THREADS_WIN32)
762     /* FIXME - NOT YET IMPLEMENTED. */
763     abort();
764 #endif
765 }
766 #endif
767
768 #endif
769 #endif
770
771 #endif
772
773 /*****\

```

```

774 *
775 \*****/
776
777 #endif
778 #else
779
780 /*****\
781 * No Threading Support.
782 \*****/
783
784 #endif
785
791 #ifdef __cplusplus
792 }
793 #endif
794
795 #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.](#)

```

1 /*
2  * Copyright (c) 2004 Michael David Adams.
3  * All rights reserved.
4  */
5
6 /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia

```



```

13  *
14  * All rights reserved.
15  *
16  * Permission is hereby granted, free of charge, to any person (the
17  * "User") obtaining a copy of this software and associated documentation
18  * files (the "Software"), to deal in the Software without restriction,
19  * including without limitation the rights to use, copy, modify, merge,
20  * publish, distribute, and/or sell copies of the Software, and to permit
21  * persons to whom the Software is furnished to do so, subject to the
22  * following conditions:
23  *
24  * 1. The above copyright notices and this permission notice (which
25  * includes the disclaimer below) shall be included in all copies or
26  * substantial portions of the Software.
27  *
28  * 2. The name of a copyright holder shall not be used to endorse or
29  * promote products derived from the Software without specific prior
30  * written permission.
31  *
32  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
43  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
48  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
50  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58  *
59  * __END_OF_JASPER_LICENSE__
60  */
61
62 #ifndef JAS_TMR_H
63 #define JAS_TMR_H
64
65 /* The configuration header file should be included first. */
66 #include <jasper/jas_config.h>
67
68 #if defined(JAS_HAVE_SYS_TIME_H)
69 #include <sys/time.h>
70 #else
71 #include <time.h>
72 #endif
73
74 #ifdef __cplusplus
75 extern "C" {
76 #endif
77
78 #if defined(JAS_HAVE_GETTIMEOFDAY)
79 typedef struct {
80     struct timeval start;
81     struct timeval stop;
82 } jas_tmr_t;
83
84 #elif defined(JAS_HAVE_GETRUSAGE)
85 typedef struct {
86     struct rusage start;
87     struct rusage stop;
88 } jas_tmr_t;
89 #else
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107

```

```

108 typedef int jas_tmr_t;
109
110 #endif
111
112 JAS_EXPORT
113 void jas_tmr_start(jas_tmr_t *tmr);
114
115 JAS_EXPORT
116 void jas_tmr_stop(jas_tmr_t *tmr);
117
118 JAS_EXPORT
119 double jas_tmr_get(jas_tmr_t *tmr);
120
121 #ifdef __cplusplus
122 }
123 #endif
124 #endif
125 #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_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.

16.37.1 Detailed Description

Tag/Value Pair Parser.

16.38 jas_tvp.h

[Go to the documentation of this file.](#)

```

1 /*
2  * Copyright (c) 2001-2002 Michael David Adams.
3  * All rights reserved.
4  */
5
6 /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
43 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
48 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
50 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58 *
59 * __END_OF_JASPER_LICENSE__
60 */
61
62 #ifndef JAS_TVP_H
63 #define JAS_TVP_H
64
65 /*****\
66 * Includes.

```

```

72 \*****/
73
74 /* The configuration header file should be included first. */
75 #include <jasper/jas_config.h>
76
77 #ifdef __cplusplus
78 extern "C" {
79 #endif
80
81 /*****\
82 * Types.
83 \*****/
84
85 typedef struct {
86     int id;
87     /* The ID for the tag. */
88     const char *name;
89     /* The name of the tag. */
90 } jas_taginfo_t;
91
92 typedef struct {
93     char *buf;
94     /* The parsing buffer. */
95     char *tag;
96     /* The current tag name. */
97     const char *val;
98     /* The current value. */
99     char *pos;
100    /* The current position in the parsing buffer. */
101 } jas_tvparser_t;
102
103 /*****\
104 * Tag information functions.
105 \*****/
106
107 JAS_ATTRIBUTE_PURE
108 JAS_EXPORT
109 const jas_taginfo_t *jas_taginfos_lookup(const jas_taginfo_t *taginfos,
110     const char *name);
111
112 JAS_ATTRIBUTE_PURE
113 JAS_EXPORT
114 const jas_taginfo_t *jas_taginfo_nonnull(const jas_taginfo_t *taginfo);
115
116 /*****\
117 * Tag-value parser functions.
118 \*****/
119
120 JAS_EXPORT
121 jas_tvparser_t *jas_tvparser_create(const char *s);
122
123 JAS_EXPORT
124 void jas_tvparser_destroy(jas_tvparser_t *tvparser);
125
126 JAS_EXPORT
127 int jas_tvparser_next(jas_tvparser_t *tvparser);
128
129 JAS_ATTRIBUTE_PURE
130 JAS_EXPORT
131 const char *jas_tvparser_gettag(const jas_tvparser_t *tvparser);
132
133 JAS_ATTRIBUTE_PURE
134 JAS_EXPORT
135 const char *jas_tvparser_getval(const jas_tvparser_t *tvparser);
136
137 #ifdef __cplusplus
138 }
139 #endif
140 #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.](#)

```
1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2003 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
```

```

46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 #ifndef JAS_TYPES_H
65 #define JAS_TYPES_H
66
67 /* The configuration header file should be included first. */
68 #include <jasper/jas_config.h>
69
70 /* Note: The immediately following header files should eventually be removed. */
71 #include <stddef.h> /* IWYU pragma: export */
72 #include <stdint.h> /* IWYU pragma: export */
73
74 #include <limits.h> /* IWYU pragma: export */
75
76 #if defined(JAS_HAVE_SYS_TYPES_H)
77 #include <sys/types.h> /* IWYU pragma: export */
78 #endif
79
80 #ifdef __cplusplus
81 extern "C" {
82 #endif
83
84 #define jas_uchar unsigned char
85 #define jas_uint unsigned int
86 #define jas_ulong unsigned long
87 #define jas_longlong long long
88 #define jas_ulonglong unsigned long long
89
90 #if !defined(JAS_NO_SET_SSIZE_T)
91 #   if !defined(SSIZE_MAX)
92 #       if (JAS_SIZEOF_INT == JAS_SIZEOF_SIZE_T)
93 #           define ssize_t int
94 #           define SSIZE_MAX INT_MAX
95 #       elif (JAS_SIZEOF_LONG == JAS_SIZEOF_SIZE_T)
96 #           define ssize_t long
97 #           define SSIZE_MAX LONG_MAX
98 #       else
99 #           define ssize_t jas_longlong
100 #           define SSIZE_MAX LLONG_MAX
101 #       endif
102 #   endif
103 #endif
104
105 #if 0
106 #if defined(JAS_HAVE_SSIZE_T)
107 #define jas_ssize_t ssize_t
108 #define JAS_SSIZE_MAX SSIZE_MAX
109 #else
110 #define jas_ssize_t jas_longlong
111 #define JAS_SSIZE_MAX LLONG_MAX
112 #endif
113 #endif
114
115 #if defined(_MSC_VER) && (_MSC_VER < 1800)
116 #define bool int
117 #define false 0
118 #define true 1
119
120 #define PRIxFAST32 "x"
121 #define PRIxFAST16 PRIxFAST32
122 #define PRIuFAST32 "u"
123 #define PRIuFAST16 PRIuFAST32
124 #define PRIiFAST32 "i"
125 #endif
126
127 #ifdef _WIN64
128 #define PRIdPTR "lld"
129 #endif

```

```

132 #else
133     #define PRIdPTR "d"
134 #endif
135
136 #ifndef _HUGE_ENUF
137     #define _HUGE_ENUF 1e+300
138 #endif
139
140 #define INFINITY ((float) (_HUGE_ENUF * _HUGE_ENUF))
141
142 #define strtoull _strtoui64
143
144 #else
145 #include <stdbool.h> /* IWYU pragma: export */
146 #include <inttypes.h> /* IWYU pragma: export */
147 #endif
148
149 /* The below macro is intended to be used for type casts. By using this
150    macro, type casts can be easily located in the source code with
151    tools like "grep". */
152 #define JAS_CAST(t, e) \
153     ((t) (e))
154
155 /* The number of bits in the integral type uint_fast32_t. */
156 /* NOTE: This could underestimate the size on some exotic architectures. */
157 #define JAS_UINTFAST32_NUMBITS (8 * sizeof(uint_fast32_t))
158
159 #if 0
160 #if defined(JAS_HAVE_MAX_ALIGN_T)
161 #define jas_max_align_t max_align_t
162 #else
163 #define jas_max_align_t long double
164 #endif
165 #endif
166
167 /*
168 Assume that a compiler claiming to be compliant with C11 or a later version
169 of the C standard provides a suitable definition of max_align_t.
170 The JAS_NO_SET_MAX_ALIGN_T preprocessor symbol can be used to override
171 this behavior.
172 */
173 #if defined(JAS_NO_SET_MAX_ALIGN_T)
174     /*
175      The user of this header is assuming responsibility for providing a
176      suitable definition for max_align_t.
177      */
178 #elif defined(_MSC_VER)
179     /*
180      Define max_align_t as a preprocessor symbol since using typedef will
181      cause problems.
182      */
183 #define max_align_t long double
184 #elif !(defined(__STDC_VERSION__) && (__STDC_VERSION__ - 0 >= 201112L))
185 #define max_align_t long double
186 #endif
187
188 #if 0
189 #if defined(JAS_HAVE_UINTMAX_T)
190 #define jas_uintmax_t uintmax_t
191 #else
192 #define jas_uintmax_t uint_fast64_t
193 #endif
194 #endif
195
196 #if 0
197 #if defined(JAS_HAVE_INTMAX_T)
198 #define jas_intmax_t intmax_t
199 #else
200 #define jas_intmax_t int_fast64_t
201 #endif
202 #endif
203
204 #ifdef __cplusplus
205 }
206 #endif
207
208 #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.](#)

```
1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2003 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
```



```

53  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60  *
61  * __END_OF_JASPER_LICENSE__
62  */
63
64 #ifndef JAS_VERSION_H
65 #define JAS_VERSION_H
66
67 /* The configuration header file should be included first. */
68 #include <jasper/jas_config.h>
69
70 #ifdef __cplusplus
71 extern "C" {
72 #endif
73
74 /*****
75  * Constants and types.
76  *****/
77
78 #define JAS_COPYRIGHT \
79     "Copyright (c) 2001-2022 Michael David Adams.\n" \
80     "Copyright (c) 1999-2000 Image Power, Inc. and the University of\n" \
81     "    British Columbia.\n" \
82     "All rights reserved.\n"
83
84 #define JAS_NOTES \
85     "For more information about this software, please visit the following\n" \
86     "web sites/pages:\n" \
87     "    https://jasper-software.github.io/jasper\n" \
88     "    https://jasper-software.github.io/jasper-manual\n" \
89     "    https://www.ece.uvic.ca/~mdadams/jasper\n" \
90     "Please submit bug reports using the bug-tracker at:\n" \
91     "    https://github.com/jasper-software/jasper/issues\n"
92
93 /*****
94  * Functions.
95  *****/
96
97 JAS_ATTRIBUTE_CONST
98 JAS_EXPORT
99 const char *jas_getversion(void);
100 /* Get the version information for the JasPer library. */
101 /* Note: Since libjasper can be built as a shared library, the version
102    returned by this function may not necessarily correspond to JAS_VERSION. */
103
104 #ifdef __cplusplus
105 }
106 #endif
107 #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.](#)

```
1 /*
2  * Copyright (c) 2001-2003 Michael David Adams.
3  * All rights reserved.
4  */
5
6 /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
43 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
```

```

47  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
48  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
50  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58  *
59  * __END_OF_JASPER_LICENSE__
60  */
61
62 #ifndef JAS_JASPER_H
63 #define JAS_JASPER_H
64
65 // IWYU pragma: begin_exports
66
67 /* The configuration header file should be included first. */
68 #include <jasper/jas_config.h>
69
70 #include <jasper/jas_types.h>
71 #include <jasper/jas_version.h>
72
73 #include <jasper/jas_init.h>
74 #include <jasper/jas_cm.h>
75 #include <jasper/jas_icc.h>
76 #include <jasper/jas_fix.h>
77 #include <jasper/jas_debug.h>
78 #include <jasper/jas_getopt.h>
79 #include <jasper/jas_image.h>
80 #include <jasper/jas_math.h>
81 #include <jasper/jas_malloc.h>
82 #include <jasper/jas_seq.h>
83 #include <jasper/jas_stream.h>
84 #include <jasper/jas_string.h>
85 #include <jasper/jas_tmr.h>
86 #include <jasper/jas_tvp.h>
87 #include <jasper/jas_thread.h>
88 #include <jasper/jas_init.h>
89 #include <jasper/jas_log.h>
90
91 #ifdef __cplusplus
92 extern "C" {
93 #endif
94
95 #ifdef __cplusplus
96 }
97 #endif
98
99 // IWYU pragma: end_exports
100
101 #endif

```

16.45 jp2_cod.h

```

1  /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8  /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation

```

```

20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 /*
65 * JP2 Library
66 *
67 * $Id$
68 */
69
70 #ifndef JP2_COD_H
71 #define JP2_COD_H
72
73 /*****\
74 * Includes.
75 \*****/
76
77 #include "jasper/jas_types.h"
78 #include "jasper/jas_stream.h"
79 #include "jasper/jas_image.h"
80
81 #include <stdio.h>
82
83 /*****\
84 * Macros.
85 \*****/
86
87 #define JP2_SPTOBPC(s, p) \
88     (((p) - 1) & 0x7f) | (((s) & 1) << 7)
89
90 /*****\
91 * Box class.
92 \*****/
93
94 #define JP2_BOX_HDRLEN(ext) ((ext) ? 16 : 8)
95
96 /* Box types. */
97 #define JP2_BOX_JP                0x6a502020    /* Signature */
98 #define JP2_BOX_FTYPE             0x66747970    /* File Type */
99 #define JP2_BOX_JP2H              0x6a703268    /* JP2 Header */
100 #define JP2_BOX_IHDR              0x69686472    /* Image Header */

```

```

101 #define JP2_BOX_BPCC      0x62706363      /* Bits Per Component */
102 #define JP2_BOX_COLR      0x636f6c72      /* Color Specification */
103 #define JP2_BOX_PCLR      0x70636c72      /* Palette */
104 #define JP2_BOX_CMAP      0x636d6170      /* Component Mapping */
105 #define JP2_BOX_CDEF      0x63646566      /* Channel Definition */
106 #define JP2_BOX_RES       0x72657320      /* Resolution */
107 #define JP2_BOX_RESC      0x72657363      /* Capture Resolution */
108 #define JP2_BOX_RESD      0x72657364      /* Default Display Resolution */
109 #define JP2_BOX_JP2C      0x6a703263      /* Contiguous Code Stream */
110 #define JP2_BOX_JP2I      0x6a703269      /* Intellectual Property */
111 #define JP2_BOX_XML       0x786d6c20      /* XML */
112 #define JP2_BOX_UUID      0x75756964      /* UUID */
113 #define JP2_BOX_UINF      0x75696e66      /* UUID Info */
114 #define JP2_BOX_ULST      0x75637374      /* UUID List */
115 #define JP2_BOX_URL       0x75726c20      /* URL */
116
117 #define JP2_BOX_SUPER      0x01
118 #define JP2_BOX_NODATA    0x02
119
120 /* JP box data. */
121
122 #define JP2_JP_MAGIC      0x0d0a870a
123 #define JP2_JP_LEN        12
124
125 typedef struct {
126     uint_fast32_t magic;
127 } jp2_jp_t;
128
129 /* Ftyp box data. */
130
131 #define JP2_Ftyp_MAXCOMPATCODES 32
132 #define JP2_Ftyp_MAJVER      0x6a703220
133 #define JP2_Ftyp_MINVER      0
134 #define JP2_Ftyp_COMPATCODE  JP2_Ftyp_MAJVER
135
136 typedef struct {
137     uint_fast32_t majver;
138     uint_fast32_t minver;
139     uint_fast32_t numcompatcodes;
140     uint_fast32_t compatcodes[JP2_Ftyp_MAXCOMPATCODES];
141 } jp2_ftyp_t;
142
143 /* IHDR box data. */
144
145 #define JP2_IHDR_COMPTYPE    7
146 #define JP2_IHDR_BPCNULL    255
147
148 typedef struct {
149     uint_fast32_t width;
150     uint_fast32_t height;
151     uint_fast16_t numcmpts;
152     uint_fast8_t bpc;
153     uint_fast8_t comptype;
154     uint_fast8_t csunk;
155     uint_fast8_t ipr;
156 } jp2_ihdr_t;
157
158 /* BPCC box data. */
159
160 typedef struct {
161     uint_fast16_t numcmpts;
162     uint_fast8_t *bpcs;
163 } jp2_bpcc_t;
164
165 /* COLR box data. */
166
167 #define JP2_COLR_ENUM      1
168 #define JP2_COLR_ICC      2
169 #define JP2_COLR_PRI      0
170
171 #define JP2_COLR_SRGB      16
172 #define JP2_COLR_SGRAY     17
173 #define JP2_COLR_SYCC      18
174
175 typedef struct {
176     uint_fast8_t method;
177     uint_fast8_t pri;
178     uint_fast8_t approx;
179     uint_fast32_t csid;
180     uint_fast8_t *iccp;
181     size_t iccplen;

```

```

182      /* XXX - Someday we ought to add ICC profile data here. */
183 } jp2_colr_t;
184
185 /* PCLR box data. */
186
187 typedef struct {
188     uint_fast16_t numlutents;
189     uint_fast8_t numchans;
190     int_fast32_t *lutdata;
191     uint_fast8_t *bpc;
192 } jp2_pclr_t;
193
194 /* CDEF box per-channel data. */
195
196 #define JP2_CDEF_RGB_R 1
197 #define JP2_CDEF_RGB_G 2
198 #define JP2_CDEF_RGB_B 3
199
200 #define JP2_CDEF_YCBCR_Y 1
201 #define JP2_CDEF_YCBCR_CB 2
202 #define JP2_CDEF_YCBCR_CR 3
203
204 #define JP2_CDEF_GRAY_Y 1
205
206 #define JP2_CDEF_TYPE_COLOR 0
207 #define JP2_CDEF_TYPE_OPACITY 1
208 #define JP2_CDEF_TYPE_UNSPEC 65535
209 #define JP2_CDEF_ASOC_ALL 0
210 #define JP2_CDEF_ASOC_NONE 65535
211
212 typedef struct {
213     uint_fast16_t channo;
214     uint_fast16_t type;
215     uint_fast16_t assoc;
216 } jp2_cdefchan_t;
217
218 /* CDEF box data. */
219
220 typedef struct {
221     uint_fast16_t numchans;
222     jp2_cdefchan_t *ents;
223 } jp2_cdef_t;
224
225 typedef struct {
226     uint_fast16_t cmptno;
227     uint_fast8_t map;
228     uint_fast8_t pcol;
229 } jp2_cmapent_t;
230
231 typedef struct {
232     uint_fast16_t numchans;
233     jp2_cmapent_t *ents;
234 } jp2_cmap_t;
235
236 #define JP2_CMAP_DIRECT 0
237 #define JP2_CMAP_PALETTE 1
238
239 /* Generic box. */
240
241 struct jp2_boxops_s;
242 typedef struct {
243
244     const struct jp2_boxops_s *ops;
245     const struct jp2_boxinfo_s *info;
246
247     uint_fast32_t type;
248
249     /* The length of the box including the (variable-length) header. */
250     uint_fast32_t len;
251
252     /* The length of the box data. */
253     uint_fast32_t datalen;
254
255     union {
256         jp2_jp_t jp;
257         jp2_ftyp_t ftyp;
258         jp2_ihdr_t ihdr;
259         jp2_bpcc_t bpcc;
260         jp2_colr_t colr;
261         jp2_pclr_t pclr;
262         jp2_cdef_t cdef;

```

```

263         jp2_cmap_t cmap;
264     } data;
265
266 } jp2_box_t;
267
268 typedef struct jp2_boxops_s {
269     void (*init)(jp2_box_t *box);
270     void (*destroy)(jp2_box_t *box);
271     int (*getdata)(jp2_box_t *box, jas_stream_t *in);
272     int (*putdata)(const jp2_box_t *box, jas_stream_t *out);
273     void (*dumpdata)(const jp2_box_t *box);
274 } jp2_boxops_t;
275
276 /*****
277 *
278 *****/
279
280 typedef struct jp2_boxinfo_s {
281     int type;
282     int flags;
283     const char *name;
284     jp2_boxops_t ops;
285 } jp2_boxinfo_t;
286
287 /*****
288 * Box class.
289 *****/
290
291 jp2_box_t *jp2_box_create(int type);
292 void jp2_box_destroy(jp2_box_t *box);
293 jp2_box_t *jp2_box_get(jas_stream_t *in);
294 int jp2_box_put(jp2_box_t *box, jas_stream_t *out);
295
296 JAS_ATTRIBUTE_CONST
297 static inline uint_least8_t JP2_DTYPETOBCP(uint_least8_t dtype)
298 {
299     return (JAS_IMAGE_CDT_GETSGND(dtype) « 7) | (JAS_IMAGE_CDT_GETPREC(dtype) - 1);
300 }
301
302 JAS_ATTRIBUTE_CONST
303 static inline uint_least8_t JP2_BPCTODTYPE(uint_least8_t bpc)
304 {
305     return JAS_IMAGE_CDT_SETSGND(bpc « 7) | JAS_IMAGE_CDT_SETPREC((bpc & 0x7f) + 1);
306 }
307
308 #define ICC_CS_RGB      0x52474220
309 #define ICC_CS_YCBCR    0x59436272
310 #define ICC_CS_GRAY     0x47524159
311
312 const jp2_cdefchan_t *jp2_cdef_lookup(jp2_cdef_t *cdef, int channo);
313
314 #endif

```

16.46 jp2_dec.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  * British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10  * JasPer License Version 2.0
11  *
12  * Copyright (c) 2001-2006 Michael David Adams
13  * Copyright (c) 1999-2000 Image Power, Inc.
14  * Copyright (c) 1999-2000 The University of British Columbia
15  *
16  * All rights reserved.
17  *
18  * Permission is hereby granted, free of charge, to any person (the
19  * "User") obtaining a copy of this software and associated documentation
20  * files (the "Software"), to deal in the Software without restriction,
21  * including without limitation the rights to use, copy, modify, merge,
22  * publish, distribute, and/or sell copies of the Software, and to permit

```

```

23  * persons to whom the Software is furnished to do so, subject to the
24  * following conditions:
25  *
26  * 1. The above copyright notices and this permission notice (which
27  * includes the disclaimer below) shall be included in all copies or
28  * substantial portions of the Software.
29  *
30  * 2. The name of a copyright holder shall not be used to endorse or
31  * promote products derived from the Software without specific prior
32  * written permission.
33  *
34  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60  *
61  * __END_OF_JASPER_LICENSE__
62  */
63
64 #ifndef JP2_DEC_H
65 #define JP2_DEC_H
66
67 #include "jasper/jas_image.h"
68 #include "jp2_cod.h"
69
70 typedef struct {
71
72     jp2_box_t *pclr;
73     jp2_box_t *cdef;
74     jp2_box_t *ihdr;
75     jp2_box_t *bpcc;
76     jp2_box_t *cmap;
77     jp2_box_t *colr;
78     jas_image_t *image;
79     uint_fast16_t numchans;
80     uint_fast16_t *chantocmplut;
81
82 } jp2_dec_t;
83
84 #endif

```

16.47 jpc_bs.h

```

1  /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8  /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams

```



```

13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 /*
65 * Bit Stream Class
66 *
67 * $Id$
68 */
69
70 #ifndef JPC_BS_H
71 #define JPC_BS_H
72
73 /*****\
74 * Includes.
75 \*****/
76
77 #include "jasper/jas_types.h"
78 #include "jasper/jas_stream.h"
79
80 #include <assert.h>
81 #include <stdio.h>
82
83 /*****\
84 * Constants.
85 \*****/
86
87 /*
88 * Bit stream open mode flags.
89 */
90
91 /* Bit stream open for reading. */
92 #define JPC_BITSTREAM_READ 0x01
93 /* Bit stream open for writing. */

```

```

94 #define JPC_BITSTREAM_WRITE      0x02
95
96 /*
97  * Bit stream flags.
98  */
99
100 /* Do not close underlying character stream. */
101 #define JPC_BITSTREAM_NOCLOSE    0x01
102 /* End of file has been reached while reading. */
103 #define JPC_BITSTREAM_EOF        0x02
104 /* An I/O error has occurred. */
105 #define JPC_BITSTREAM_ERR        0x04
106
107 /*****
108  * Types.
109  *****/
110
111 /* Bit stream class. */
112
113 typedef struct {
114     /* Some miscellaneous flags. */
115     int flags_;
116
117     /* The input/output buffer. */
118     uint_fast16_t buf_;
119
120     /* The number of bits remaining in the byte being read/written. */
121     int cnt_;
122
123     /* The underlying stream associated with this bit stream. */
124     jas_stream_t *stream_;
125
126     /* The mode in which this bit stream was opened. */
127     int openmode_;
128 } jpc_bitstream_t;
129
130 /*****
131  * Functions/macros for opening and closing bit streams..
132  *****/
133
134 /* Open a stream as a bit stream. */
135 jpc_bitstream_t *jpc_bitstream_sopen(jas_stream_t *stream, const char *mode);
136
137 /* Close a bit stream. */
138 int jpc_bitstream_close(jpc_bitstream_t *bitstream);
139
140 /*****
141  * Functions/macros for reading from and writing to bit streams..
142  *****/
143
144 /* Read a bit from a bit stream. */
145 #ifndef NDEBUG
146 #define jpc_bitstream_getbit(bitstream) \
147     jpc_bitstream_getbit_func(bitstream)
148 #else
149 #define jpc_bitstream_getbit(bitstream) \
150     jpc_bitstream_getbit_macro(bitstream)
151 #endif
152
153 /* Write a bit to a bit stream. */
154 #ifndef NDEBUG
155 #define jpc_bitstream_putbit(bitstream, v) \
156     jpc_bitstream_putbit_func(bitstream, v)
157 #else
158 #define jpc_bitstream_putbit(bitstream, v) \
159     jpc_bitstream_putbit_macro(bitstream, v)
160 #endif
161
162 /* Read one or more bits from a bit stream. */
163 long jpc_bitstream_getbits(jpc_bitstream_t *bitstream, int n);
164
165 /* Write one or more bits to a bit stream. */
166 int jpc_bitstream_putbits(jpc_bitstream_t *bitstream, int n, long v);
167
168 /*****
169  * Functions/macros for flushing and aligning bit streams.
170  *****/
171
172 /* Align the current position within the bit stream to the next byte

```

```

175     boundary. */
176 int jpc_bitstream_align(jpc_bitstream_t *bitstream);
177
178 /* Align the current position in the bit stream with the next byte boundary,
179    ensuring that certain bits consumed in the process match a particular
180    pattern. */
181 int jpc_bitstream_inalign(jpc_bitstream_t *bitstream, int fillmask,
182    int filldata);
183
184 /* Align the current position in the bit stream with the next byte boundary,
185    writing bits from the specified pattern (if necessary) in the process. */
186 int jpc_bitstream_outalign(jpc_bitstream_t *bitstream, int filldata);
187
188 /* Check if a bit stream needs alignment. */
189 JAS_ATTRIBUTE_PURE
190 int jpc_bitstream_needalign(const jpc_bitstream_t *bitstream);
191
192 /* How many additional bytes would be output if the bit stream was aligned? */
193 JAS_ATTRIBUTE_PURE
194 int jpc_bitstream_pending(const jpc_bitstream_t *bitstream);
195
196 /*****
197  * Functions/macros for querying state information for bit streams.
198  *****/
199
200 /* Has EOF been encountered on a bit stream? */
201 #define jpc_bitstream_eof(bitstream) \
202     ((bitstream)->flags_ & JPC_BITSTREAM_EOF)
203
204 /*****
205  * Internals.
206  *****/
207
208 /* DO NOT DIRECTLY INVOKE ANY OF THE MACROS OR FUNCTIONS BELOW.  THEY ARE
209    FOR INTERNAL USE ONLY. */
210
211 int jpc_bitstream_getbit_func(jpc_bitstream_t *bitstream);
212
213 int jpc_bitstream_putbit_func(jpc_bitstream_t *bitstream, int v);
214
215 int jpc_bitstream_fillbuf(jpc_bitstream_t *bitstream);
216
217 #define jpc_bitstream_getbit_macro(bitstream) \
218     (assert((bitstream)->openmode_ & JPC_BITSTREAM_READ), \
219      (--(bitstream)->cnt_ >= 0) ? \
220      ((int)((bitstream)->buf_ » (bitstream)->cnt_) & 1) : \
221      jpc_bitstream_fillbuf(bitstream))
222
223 #define jpc_bitstream_putbit_macro(bitstream, bit) \
224     (assert((bitstream)->openmode_ & JPC_BITSTREAM_WRITE), \
225      (--(bitstream)->cnt_ < 0) ? \
226      ((bitstream)->buf_ = ((bitstream)->buf_ « 8) & 0xffff, \
227       (bitstream)->cnt_ = ((bitstream)->buf_ == 0xff00) ? 6 : 7, \
228       (bitstream)->buf_ |= ((bit) & 1) « (bitstream)->cnt_, \
229       (jas_stream_putc((bitstream)->stream_, (bitstream)->buf_ » 8) == EOF) \
230       ? (EOF) : ((bit) & 1)) : \
231      ((bitstream)->buf_ |= ((bit) & 1) « (bitstream)->cnt_, \
232       (bit) & 1))
233
234 #endif

```

16.48 jpc_cod.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia

```

```

15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 /*
65 * $Id$
66 */
67
68 #ifndef JPC_COD_H
69 #define JPC_COD_H
70
71 #include "jpc_tlcod.h"
72
73 /*****\
74 * Constants.
75 \*****/
76
77 /* The nominal word size used by this implementation. */
78 #define JPC_PREC 32
79
80 void jpc_init(void);
81
82 #endif

```

16.49 jpc_cs.h

```

1 /*
2 * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3 * British Columbia.
4 * Copyright (c) 2001-2002 Michael David Adams.
5 * All rights reserved.
6 */

```

```

7
8 /* __START_OF_JASPER_LICENSE__
9 *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 /*
65 * JPEG-2000 Code Stream Library
66 *
67 * $Id$
68 */
69
70 #ifndef JPC_CS_H
71 #define JPC_CS_H
72
73 /*****\
74 * Includes.
75 \*****/
76
77 #include "jasper/jas_stream.h"
78
79 #include <assert.h>
80 #include <stdio.h>
81
82 /*****\
83 * Constants and Types.
84 \*****/
85
86 /* The maximum number of resolution levels. */
87 #define JPC_MAXRLVLS 33

```

```

88
89 /* The maximum number of bands. */
90 #define JPC_MAXBANDS      (3 * JPC_MAXRLVLS + 1)
91
92 /* The maximum number of layers. */
93 #define JPC_MAXLYRS      16384
94
95 /*****\
96 * Code stream.
97 \*****/
98
99 /*
100  * Code stream states.
101  */
102
103 /* Initial. */
104 #define JPC_CS_INIT      0
105 /* Main header. */
106 #define JPC_CS_MHDR      1
107 /* Tile-part header. */
108 #define JPC_CS_THDR      2
109 /* Main trailer. */
110 #define JPC_CS_MTLR      3
111 /* Tile-part data. */
112 #define JPC_CS_TDATA      4
113
114 /*
115  * Unfortunately, the code stream syntax was not designed in such a way that
116  * any given marker segment can be correctly decoded without additional state
117  * derived from previously decoded marker segments.
118  * For example, a RGN/COC/QCC marker segment cannot be decoded unless the
119  * number of components is known.
120  */
121
122 /*
123  * Code stream state information.
124  */
125
126 typedef struct {
127
128     /* The number of components. */
129     uint_fast16_t numcomps;
130
131 } jpc_cstate_t;
132
133 /*****\
134 * SOT marker segment parameters.
135 \*****/
136
137 typedef struct {
138
139     /* The tile number. */
140     uint_fast16_t tileno;
141
142     /* The combined length of the marker segment and its auxiliary data
143      (i.e., packet data). */
144     uint_fast32_t len;
145
146     /* The tile-part instance. */
147     uint_fast8_t partno;
148
149     /* The number of tile-parts. */
150     uint_fast8_t numparts;
151
152 } jpc_sot_t;
153
154 /*****\
155 * SIZ marker segment parameters.
156 \*****/
157
158 /* Per component information. */
159
160 typedef struct {
161
162     /* The precision of the samples. */
163     uint_fast8_t prec;
164
165     /* The signedness of the samples. */
166     uint_fast8_t sgnd;
167
168     /* The horizontal separation of samples with respect to the reference

```

```

169         grid. */
170         uint_fast8_t hsamp;
171
172         /* The vertical separation of samples with respect to the reference
173         grid. */
174         uint_fast8_t vsamp;
175
176     } jpc_sizcomp_t;
177
178     /* SIZ marker segment parameters. */
179
180     typedef struct {
181
182         /* The code stream capabilities. */
183         uint_fast16_t caps;
184
185         /* The width of the image in units of the reference grid. */
186         uint_fast32_t width;
187
188         /* The height of the image in units of the reference grid. */
189         uint_fast32_t height;
190
191         /* The horizontal offset from the origin of the reference grid to the
192         left side of the image area. */
193         uint_fast32_t xoff;
194
195         /* The vertical offset from the origin of the reference grid to the
196         top side of the image area. */
197         uint_fast32_t yoff;
198
199         /* The nominal width of a tile in units of the reference grid. */
200         uint_fast32_t tilewidth;
201
202         /* The nominal height of a tile in units of the reference grid. */
203         uint_fast32_t tileheight;
204
205         /* The horizontal offset from the origin of the reference grid to the
206         left side of the first tile. */
207         uint_fast32_t tilexoff;
208
209         /* The vertical offset from the origin of the reference grid to the
210         top side of the first tile. */
211         uint_fast32_t tileyoff;
212
213         /* The number of components. */
214         uint_fast16_t numcomps;
215
216         /* The per-component information. */
217         jpc_sizcomp_t *comps;
218
219     } jpc_siz_t;
220
221     /* *****\
222     * COD marker segment parameters.
223     \***** */
224
225     /*
226     * Coding style constants.
227     */
228
229     /* Precincts may be used. */
230     #define JPC_COX_PRT      0x01
231     /* SOP marker segments may be used. */
232     #define JPC_COD_SOP      0x02
233     /* EPH marker segments may be used. */
234     #define JPC_COD_EPH      0x04
235
236     /*
237     * Progression order constants.
238     */
239
240     /* Layer-resolution-component-precinct progressive
241     (i.e., progressive by fidelity). */
242     #define JPC_COD_LRCPPRG 0
243     /* Resolution-layer-component-precinct progressive
244     (i.e., progressive by resolution). */
245     #define JPC_COD_RLCPPRG 1
246     /* Resolution-precinct-component-layer progressive. */
247     #define JPC_COD_RPCLPRG 2
248     /* Precinct-component-resolution-layer progressive. */
249     #define JPC_COD_PCRLPRG 3

```

```

250 /* Component-position-resolution-layer progressive. */
251 #define JPC_COD_CPRLPRG 4
252
253 /*
254  * Code block style constants.
255  */
256
257 #define JPC_COX_LAZY    0x01 /* Selective arithmetic coding bypass. */
258 #define JPC_COX_RESET  0x02 /* Reset context probabilities. */
259 #define JPC_COX_TERMALL 0x04 /* Terminate all coding passes. */
260 #define JPC_COX_VSC     0x08 /* Vertical stripe causal context formation. */
261 #define JPC_COX_PTERM   0x10 /* Predictable termination. */
262 #define JPC_COX_SEGSYM  0x20 /* Use segmentation symbols. */
263
264 /* Transform constants. */
265 #define JPC_COX_INS     0x00 /* Irreversible 9/7. */
266 #define JPC_COX_RFT     0x01 /* Reversible 5/3. */
267
268 /* Multicomponent transform constants. */
269 #define JPC_COD_NOMCT   0x00 /* No multicomponent transform. */
270 #define JPC_COD_MCT     0x01 /* Multicomponent transform. */
271
272 /* Get the code block size value from the code block size exponent. */
273 JAS_ATTRIBUTE_CONST
274 static inline unsigned JPC_COX_CBLKSIZEEXPN(unsigned x)
275 {
276     return x - 2;
277 }
278
279 /* Get the code block size exponent from the code block size value. */
280 JAS_ATTRIBUTE_CONST
281 static inline unsigned JPC_COX_GETCBLKSIZEEXPN(unsigned x)
282 {
283     return x + 2;
284 }
285
286 /* Per resolution-level information. */
287
288 typedef struct {
289     /* The packet partition width. */
290     uint_fast8_t parwidthval;
291
292     /* The packet partition height. */
293     uint_fast8_t parheightval;
294 } jpc_coxrlvl_t;
295
296 /* Per component information. */
297
298 typedef struct {
299     /* The coding style. */
300     uint_fast8_t csty;
301
302     /* The number of decomposition levels. */
303     uint_fast8_t numdlvls;
304
305     /* The nominal code block width specifier. */
306     uint_fast8_t cblkwidthval;
307
308     /* The nominal code block height specifier. */
309     uint_fast8_t cblkheightval;
310
311     /* The style of coding passes. */
312     uint_fast8_t cblksty;
313
314     /* The QMFB employed. */
315     uint_fast8_t qmfbid;
316
317     /* The number of resolution levels. */
318     int numrlvls;
319
320     /* The per-resolution-level information. */
321     jpc_coxrlvl_t rlvls[JPC_MAXRLVLS];
322 } jpc_coxcp_t;
323
324 /* COD marker segment parameters. */
325
326 typedef struct {

```



```

331
332     /* The general coding style. */
333     uint_fast8_t csty;
334
335     /* The progression order. */
336     uint_fast8_t prg;
337
338     /* The number of layers. */
339     uint_fast16_t numlyrs;
340
341     /* The multicomponent transform. */
342     uint_fast8_t mctrans;
343
344     /* Component-related parameters. */
345     jpc_coxcp_t compparms;
346
347 } jpc_cod_t;
348
349 /* COC marker segment parameters. */
350
351 typedef struct {
352
353     /* The component number. */
354     uint_fast16_t compno;
355
356     /* Component-related parameters. */
357     jpc_coxcp_t compparms;
358
359 } jpc_coc_t;
360
361 /*****\
362  * RGN marker segment parameters.
363  \*****/
364
365 /* The maxshift ROI style. */
366 #define JPC_RGN_MAXSHIFT      0x00
367
368 typedef struct {
369
370     /* The component to which the marker applies. */
371     uint_fast16_t compno;
372
373     /* The ROI style. */
374     uint_fast8_t roisty;
375
376     /* The ROI shift value. */
377     uint_fast8_t roishift;
378
379 } jpc_rgn_t;
380
381 /*****\
382  * QCD/QCC marker segment parameters.
383  \*****/
384
385 /*
386  * Quantization style constants.
387  */
388
389 #define JPC_QCX_NOQNT      0 /* No quantization. */
390 #define JPC_QCX_SIQNT      1 /* Scalar quantization, implicit. */
391 #define JPC_QCX_SEQNT      2 /* Scalar quantization, explicit. */
392
393 /*
394  * Stepsize manipulation macros.
395  */
396
397 JAS_ATTRIBUTE_CONST
398 static inline unsigned JPC_QCX_GETEXPN(unsigned x)
399 {
400     return x >> 11;
401 }
402
403 JAS_ATTRIBUTE_CONST
404 static inline unsigned JPC_QCX_GETMANT(unsigned x)
405 {
406     return x & 0x7ff;
407 }
408
409 JAS_ATTRIBUTE_CONST
410 static inline uint_fast16_t JPC_QCX_EXPNT(unsigned x)
411 {

```

```

412     assert(!(x & (~0x1f)));
413
414     return (x & 0x1f) << 11;
415 }
416
417 JAS_ATTRIBUTE_CONST
418 static inline uint_fast16_t JPC_QCX_MANT(unsigned x)
419 {
420     assert(!(x & (~0x7ff)));
421
422     return x & 0x7ff;
423 }
424
425 /* Per component information. */
426
427 typedef struct {
428
429     /* The quantization style. */
430     uint_fast8_t qntsty;
431
432     /* The number of step sizes. */
433     int numstepsizes;
434
435     /* The step sizes. */
436     uint_fast16_t *stepsizes;
437
438     /* The number of guard bits. */
439     uint_fast8_t numguard;
440 } jpc_qcxcp_t;
441
442 /* QCC marker segment parameters. */
443
444 typedef struct {
445
446     /* The component associated with this marker segment. */
447     uint_fast16_t compno;
448
449     /* The parameters. */
450     jpc_qcxcp_t compparms;
451 } jpc_qcc_t;
452
453 /* QCD marker segment parameters. */
454
455 typedef struct {
456
457     /* The parameters. */
458     jpc_qcxcp_t compparms;
459 } jpc_qcd_t;
460
461 /* *****\
462  * POD marker segment parameters.
463  \***** */
464
465 typedef struct {
466
467     /* The progression order. */
468     uint_fast8_t prgord;
469
470     /* The lower bound (inclusive) on the resolution level for the
471        progression order volume. */
472     uint_fast8_t rlvlnostart;
473
474     /* The upper bound (exclusive) on the resolution level for the
475        progression order volume. */
476     uint_fast8_t rlvlnoend;
477
478     /* The lower bound (inclusive) on the component for the progression
479        order volume. */
480     uint_fast16_t compnostart;
481
482     /* The upper bound (exclusive) on the component for the progression
483        order volume. */
484     uint_fast16_t compnoend;
485
486     /* The upper bound (exclusive) on the layer for the progression
487        order volume. */
488     uint_fast16_t lyrnoend;
489 }
490
491
492

```

```

493 } jpc_pocpchg_t;
494
495 /* An alias for the above type. */
496 typedef jpc_pocpchg_t jpc_pchg_t;
497
498 /* POC marker segment parameters. */
499
500 typedef struct {
501
502     /* The number of progression order changes. */
503     int numpchgs;
504
505     /* The per-progression-order-change information. */
506     jpc_pocpchg_t *pchgs;
507 } jpc_poc_t;
508
509 /*****\
510 * PPM/PPT marker segment parameters.
511 \*****/
512
513 /* PPM marker segment parameters. */
514
515 typedef struct {
516
517     /* The index. */
518     uint_fast8_t ind;
519
520     /* The length. */
521     uint_fast16_t len;
522
523     /* The data. */
524     jas_uchar *data;
525 } jpc_ppm_t;
526
527 /* PPT marker segment parameters. */
528
529 typedef struct {
530
531     /* The index. */
532     uint_fast8_t ind;
533
534     /* The length. */
535     uint_fast32_t len;
536
537     /* The data. */
538     unsigned char *data;
539 } jpc_ppt_t;
540
541 /*****\
542 * COM marker segment parameters.
543 \*****/
544
545 /*
546  * Registration IDs.
547  */
548
549 #define JPC_COM_BIN          0x00
550 #define JPC_COM_LATIN      0x01
551
552 typedef struct {
553
554     /* The registration ID. */
555     uint_fast16_t regid;
556
557     /* The length of the data in bytes. */
558     uint_fast16_t len;
559
560     /* The data. */
561     jas_uchar *data;
562 } jpc_com_t;
563
564 /*****\
565 * SOP marker segment parameters.
566 \*****/
567
568 typedef struct {
569

```

```

574      /* The sequence number. */
575      uint_fast16_t seqno;
576
577 } jpc_sop_t;
578
579 /*****\
580 * CRG marker segment parameters.
581 \*****/
582
583 /* Per component information. */
584
585 typedef struct {
586
587     /* The horizontal offset. */
588     uint_fast16_t hoff;
589
590     /* The vertical offset. */
591     uint_fast16_t voff;
592
593 } jpc_crgcomp_t;
594
595 typedef struct {
596
597     /* The number of components. */
598     int numcomps;
599
600     /* Per component information. */
601     jpc_crgcomp_t *comps;
602
603 } jpc_crg_t;
604
605 /*****\
606 * Marker segment parameters for unknown marker type.
607 \*****/
608
609 typedef struct {
610
611     /* The data. */
612     jas_uchar *data;
613
614     /* The length. */
615     uint_fast16_t len;
616
617 } jpc_unk_t;
618
619 /*****\
620 * Generic marker segment parameters.
621 \*****/
622
623 typedef union {
624     int soc;          /* unused */
625     jpc_sot_t sot;
626     int sod;          /* unused */
627     int eoc;          /* unused */
628     jpc_siz_t siz;
629     jpc_cod_t cod;
630     jpc_coc_t coc;
631     jpc_rgn_t rgn;
632     jpc_qcd_t qcd;
633     jpc_qcc_t qcc;
634     jpc_poc_t poc;
635     /* jpc_plm_t plm; */
636     /* jpc_plt_t plt; */
637     jpc_ppm_t ppm;
638     jpc_ppt_t ppt;
639     jpc_sop_t sop;
640     int eph;          /* unused */
641     jpc_com_t com;
642     jpc_crg_t crg;
643     jpc_unk_t unk;
644 } jpc_msparms_t;
645
646 /*****\
647 * Marker segment.
648 \*****/
649
650 /* Marker segment IDs. */
651
652 /* The smallest valid marker value. */
653 #define JPC_MS_MIN      0xff00
654

```

```

655 /* The largest valid marker value. */
656 #define JPC_MS_MAX      0xffff
657
658 /* The minimum marker value that cannot occur within packet data. */
659 #define JPC_MS_INMIN    0xff80
660 /* The maximum marker value that cannot occur within packet data. */
661 #define JPC_MS_INMAX    0xffff
662
663 /* Delimiting marker segments. */
664 #define JPC_MS_SOC      0xff4f /* Start of code stream (SOC). */
665 #define JPC_MS_SOT      0xff90 /* Start of tile-part (SOT). */
666 #define JPC_MS_SOD      0xff93 /* Start of data (SOD). */
667 #define JPC_MS_EOC      0xffd9 /* End of code stream (EOC). */
668
669 /* Fixed information marker segments. */
670 #define JPC_MS_SIZ      0xff51 /* Image and tile size (SIZ). */
671
672 /* Functional marker segments. */
673 #define JPC_MS_COD      0xff52 /* Coding style default (COD). */
674 #define JPC_MS_COC      0xff53 /* Coding style component (COC). */
675 #define JPC_MS_RGN      0xff5e /* Region of interest (RGN). */
676 #define JPC_MS_QCD      0xff5c /* Quantization default (QCD). */
677 #define JPC_MS_QCC      0xff5d /* Quantization component (QCC). */
678 #define JPC_MS_POC      0xff5f /* Progression order default (POC). */
679
680 /* Pointer marker segments. */
681 #define JPC_MS_TLM      0xff55 /* Tile-part lengths, main header (TLM). */
682 #define JPC_MS_PLM      0xff57 /* Packet length, main header (PLM). */
683 #define JPC_MS_PLT      0xff58 /* Packet length, tile-part header (PLT). */
684 #define JPC_MS_PPM      0xff60 /* Packed packet headers, main header (PPM). */
685 #define JPC_MS_PPT      0xff61 /* Packet packet headers, tile-part header (PPT). */
686
687 /* In bit stream marker segments. */
688 #define JPC_MS_SOP      0xff91 /* Start of packet (SOP). */
689 #define JPC_MS_EPH      0xff92 /* End of packet header (EPH). */
690
691 /* Informational marker segments. */
692 #define JPC_MS_CRG      0xff63 /* Component registration (CRG). */
693 #define JPC_MS_COM      0xff64 /* Comment (COM). */
694
695 /* Forward declaration. */
696 struct jpc_msops_s;
697
698 /* Generic marker segment class. */
699
700 typedef struct {
701     /* The type of marker segment. */
702     uint_fast16_t id;
703
704     /* The length of the marker segment. */
705     uint_fast16_t len;
706
707     /* The starting offset within the stream. */
708     uint_fast32_t off;
709
710     /* The parameters of the marker segment. */
711     jpc_msparams_t parms;
712
713     /* The marker segment operations. */
714     const struct jpc_msops_s *ops;
715 } jpc_ms_t;
716
717 /* Marker segment operations (which depend on the marker segment type). */
718
719 typedef struct jpc_msops_s {
720     /* Destroy the marker segment parameters. */
721     void (*destroyparms)(jpc_ms_t *ms);
722
723     /* Get the marker segment parameters from a stream. */
724     int (*getparms)(jpc_ms_t *ms, jpc_cstate_t *cstate, jas_stream_t *in);
725
726     /* Put the marker segment parameters to a stream. */
727     int (*putparms)(jpc_ms_t *ms, jpc_cstate_t *cstate, jas_stream_t *out);
728
729     /* Dump the marker segment parameters (for debugging). */
730     int (*dumpparms)(jpc_ms_t *ms);
731 } jpc_msops_t;

```

```

736
737 /*****
738  * Macros/Functions.
739  *****/
740
741 /* Create a code-stream state object. */
742 jpc_cstate_t *jpc_cstate_create(void);
743
744 /* Destroy a code-stream state object. */
745 void jpc_cstate_destroy(jpc_cstate_t *cstate);
746
747 /* Create a marker segment. */
748 jpc_ms_t *jpc_ms_create(int type);
749
750 /* Destroy a marker segment. */
751 void jpc_ms_destroy(jpc_ms_t *ms);
752
753 /* Does a marker segment have parameters? */
754 JAS_ATTRIBUTE_CONST
755 static inline bool JPC_MS_HASPARMS(unsigned x)
756 {
757     return !(x == JPC_MS_SOC || x == JPC_MS_SOD || x == JPC_MS_EOC ||
758             x == JPC_MS_EPH || (x >= 0xff30 && x <= 0xff3f));
759 }
760
761 /* Get the marker segment type. */
762 JAS_ATTRIBUTE_PURE
763 static inline unsigned jpc_ms_gettype(const jpc_ms_t *ms)
764 {
765     return ms->id;
766 }
767
768 /* Read a marker segment from a stream. */
769 jpc_ms_t *jpc_getms(jas_stream_t *in, jpc_cstate_t *cstate);
770
771 /* Write a marker segment to a stream. */
772 int jpc_putms(jas_stream_t *out, jpc_cstate_t *cstate, jpc_ms_t *ms);
773
774 /* Copy code stream data from one stream to another. */
775 int jpc_getdata(jas_stream_t *in, jas_stream_t *out, long n);
776
777 /* Copy code stream data from one stream to another. */
778 int jpc_putdata(jas_stream_t *out, jas_stream_t *in, long n);
779
780 /* Dump a marker segment (for debugging). */
781 void jpc_ms_dump(jpc_ms_t *ms);
782
783 /* Read a 8-bit unsigned integer from a stream. */
784 int jpc_getuint8(jas_stream_t *in, uint_fast8_t *val);
785
786 /* Read a 16-bit unsigned integer from a stream. */
787 int jpc_getuint16(jas_stream_t *in, uint_fast16_t *val);
788
789 /* Read a 32-bit unsigned integer from a stream. */
790 int jpc_getuint32(jas_stream_t *in, uint_fast32_t *val);
791
792 /* Write a 8-bit unsigned integer to a stream. */
793 int jpc_putuint8(jas_stream_t *out, uint_fast8_t val);
794
795 /* Write a 16-bit unsigned integer to a stream. */
796 int jpc_putuint16(jas_stream_t *out, uint_fast16_t val);
797
798 /* Write a 32-bit unsigned integer to a stream. */
799 int jpc_putuint32(jas_stream_t *out, uint_fast32_t val);
800
801 #endif

```

16.50 jpc_dec.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__

```

```

9  *
10 * Jasper License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 /*
65  * JPEG-2000 Decoder
66  *
67  * $Id$
68  */
69
70 #ifndef JPC_DEC_H
71 #define JPC_DEC_H
72
73 /*****\
74  * Includes.
75  \*****/
76
77 #include "jasper/jas_stream.h"
78 #include "jasper/jas_image.h"
79
80 #include "jpc_cod.h"
81 #include "jpc_tsfb.h"
82 #include "jpc_tagtree.h"
83 #include "jpc_cs.h"
84 #include "jpc_tlcod.h"
85 #include "jpc_t2cod.h"
86
87 /*****\
88  * Below are some ugly warts necessary to support packed packet headers.
89  \*****/

```

```

90
91 /* PPM/PPT marker segment table entry. */
92
93 typedef struct {
94
95     /* The index for this entry. */
96     uint_fast16_t ind;
97
98     /* The data length. */
99     uint_fast32_t len;
100
101     /* The data. */
102     jas_uchar *data;
103 } jpc_ppxstabent_t;
104
105 /* PPM/PPT marker segment table. */
106
107 typedef struct {
108
109     /* The number of entries. */
110     unsigned numents;
111
112     /* The maximum number of entries (i.e., the allocated size of the array
113        below). */
114     unsigned maxents;
115
116     /* The table entries. */
117     jpc_ppxstabent_t **ents;
118 } jpc_ppxstab_t;
119
120 /* Stream list class. */
121
122 typedef struct {
123
124     /* The number of streams in this list. */
125     unsigned numstreams;
126
127     /* The maximum number of streams that can be accomodated without
128        growing the streams array. */
129     unsigned maxstreams;
130
131     /* The streams. */
132     jas_stream_t **streams;
133 } jpc_streamlist_t;
134
135
136
137
138 /*****\
139 * Coding parameters class.
140 \*****/
141
142 /* Per-component coding parameters. */
143
144 typedef struct {
145
146     /* How were various coding parameters set? */
147     unsigned flags;
148
149     /* Per-component coding style parameters (e.g., explicit precinct sizes) */
150     uint_fast8_t csty;
151
152     /* The number of resolution levels. */
153     uint_fast8_t numrlvls;
154
155     /* The code block width exponent. */
156     uint_fast8_t cblkwidthexpn;
157
158     /* The code block height exponent. */
159     uint_fast8_t cblkheightexpn;
160
161     /* The QMFB ID. */
162     uint_fast8_t qmfbid;
163
164     /* The quantization style. */
165     uint_fast8_t qsty;
166
167     /* The number of quantizer step sizes. */
168     uint_fast16_t numstepsizes;
169
170     /* The step sizes. */

```



```

171     uint_fast16_t stepsizes[3 * JPC_MAXRLVLS + 1];
172
173     /* The number of guard bits. */
174     uint_fast8_t numguardbits;
175
176     /* The ROI shift value. */
177     uint_fast8_t roishift;
178
179     /* The code block parameters. */
180     uint_fast8_t cblkctx;
181
182     /* The precinct width exponents. */
183     uint_fast8_t prcwidthexpns[JPC_MAXRLVLS];
184
185     /* The precinct height exponents. */
186     uint_fast8_t prcheightexpns[JPC_MAXRLVLS];
187
188 } jpc_dec_ccp_t;
189
190 /* Coding parameters. */
191
192 typedef struct {
193
194     /* How were these coding parameters set? */
195     unsigned flags;
196
197     /* Progression change list. */
198     jpc_pchglst_t *pchglst;
199
200     /* Progression order. */
201     uint_fast8_t prgord;
202
203     /* The number of layers. */
204     uint_fast16_t numlyrs;
205
206     /* The MCT ID. */
207     uint_fast8_t mctid;
208
209     /* The coding style parameters (e.g., SOP, EPH). */
210     uint_fast8_t csty;
211
212     /* The number of components. */
213     unsigned numcomps;
214
215     /* The per-component coding parameters. */
216     jpc_dec_ccp_t *ccps;
217
218 } jpc_dec_cp_t;
219
220 /*****
221  * Decoder class.
222  *****/
223
224 /* Decoder per-segment state information. */
225
226 typedef struct jpc_dec_seg_s {
227
228     /* The next segment in the list. */
229     struct jpc_dec_seg_s *next;
230
231     /* The previous segment in the list. */
232     struct jpc_dec_seg_s *prev;
233
234     /* The starting pass number for this segment. */
235     unsigned passno;
236
237     /* The number of passes in this segment. */
238     unsigned numpasses;
239
240     /* The maximum number of passes in this segment. */
241     unsigned maxpasses;
242
243     /* The type of data in this segment (i.e., MQ or raw). */
244     enum jpc_segtype type;
245
246     /* A stream containing the data for this segment. */
247     jas_stream_t *stream;
248
249     /* The number of bytes destined for this segment from the packet
250        currently being decoded. */
251     unsigned cnt;

```

```

252
253     /* A flag indicating if this segment has been terminated. */
254     int complete;
255
256     /* The layer number to which this segment belongs. */
257     /* If the segment spans multiple layers, then the largest layer number
258        spanned by the segment is used. */
259     unsigned lyrno;
260
261 } jpc_dec_seg_t;
262
263 /* Decoder segment list. */
264
265 typedef struct {
266
267     /* The first entry in the list. */
268     jpc_dec_seg_t *head;
269
270     /* The last entry in the list. */
271     jpc_dec_seg_t *tail;
272
273 } jpc_dec_seglist_t;
274
275 /* Decoder per-code-block state information. */
276
277 typedef struct {
278
279     /* The number of passes. */
280     unsigned numpasses;
281
282     /* A list of segments that still need to be decoded. */
283     jpc_dec_seglist_t segs;
284
285     /* The first incomplete/partial segment. */
286     jpc_dec_seg_t *curseg;
287
288     /* The number of leading insignificant bit planes for this code block. */
289     unsigned numimsbs;
290
291     /* The number of bits used to encode pass data lengths. */
292     unsigned numlenbits;
293
294     /* The first pass number containing data for this code block. */
295     unsigned firstpassno;
296
297     /* The sample data associated with this code block. */
298     jas_matrix_t *data;
299
300 } jpc_dec_cblk_t;
301
302 /* Decoder per-code-block-group state information. */
303
304 typedef struct {
305
306     /* The x-coordinate of the top-left corner of the precinct. */
307     uint_fast32_t xstart;
308
309     /* The y-coordinate of the top-left corner of the precinct. */
310     uint_fast32_t ystart;
311
312     /* The x-coordinate of the bottom-right corner of the precinct
313        (plus one). */
314     uint_fast32_t xend;
315
316     /* The y-coordinate of the bottom-right corner of the precinct
317        (plus one). */
318     uint_fast32_t yend;
319
320     /* The number of code blocks spanning this precinct in the horizontal
321        direction. */
322     unsigned numhcbks;
323
324     /* The number of code blocks spanning this precinct in the vertical
325        direction. */
326     unsigned numvcbks;
327
328     /* The total number of code blocks in this precinct. */
329     unsigned numcbks;
330
331     /* The per code block information. */
332     jpc_dec_cblk_t *cblks;

```

```

333
334     /* The inclusion tag tree. */
335     jpc_tagtree_t *incltagtree;
336
337     /* The insignificant MSBs tag tree. */
338     jpc_tagtree_t *numimbsbstagtree;
339
340 } jpc_dec_prc_t;
341
342 /* Decoder per-band state information. */
343
344 typedef struct {
345
346     /* The per-code-block-group state information. */
347     jpc_dec_prc_t *prcs;
348
349     /* The sample data associated with this band. */
350     jas_matrix_t *data;
351
352     /* The orientation of this band (i.e., LL, LH, HL, or HH). */
353     enum jpc_tsfb_orient orient;
354
355     /* The encoded quantizer step size. */
356     unsigned stepsize;
357
358     /* The absolute quantizer step size. */
359     jpc_fix_t absstepsize;
360
361     /* The number of bit planes for this band. */
362     unsigned numbps;
363
364     /* The analysis gain associated with this band. */
365     int analgain;
366
367     /* The ROI shift value for this band. */
368     int roishift;
369
370 } jpc_dec_band_t;
371
372 /* Decoder per-resolution-level state information. */
373
374 typedef struct {
375
376     /* The number of bands associated with this resolution level. */
377     unsigned numbands;
378
379     /* The per-band information. */
380     jpc_dec_band_t *bands;
381
382     /* The x-coordinate of the top-left corner of the tile-component
383        at this resolution. */
384     uint_fast32_t xstart;
385
386     /* The y-coordinate of the top-left corner of the tile-component
387        at this resolution. */
388     uint_fast32_t ystart;
389
390     /* The x-coordinate of the bottom-right corner of the tile-component
391        at this resolution (plus one). */
392     uint_fast32_t xend;
393
394     /* The y-coordinate of the bottom-right corner of the tile-component
395        at this resolution (plus one). */
396     uint_fast32_t yend;
397
398     /* The exponent value for the nominal precinct width measured
399        relative to the associated LL band. */
400     unsigned prcwidthexpn;
401
402     /* The exponent value for the nominal precinct height measured
403        relative to the associated LL band. */
404     unsigned prcheightexpn;
405
406     /* The number of precincts in the horizontal direction. */
407     unsigned numhprcs;
408
409     /* The number of precincts in the vertical direction. */
410     unsigned numvprcs;
411
412     /* The total number of precincts. */
413     unsigned numprcs;

```

```

414
415     /* The exponent value for the nominal code block group width.
416        This quantity is associated with the next lower resolution level
417        (assuming that there is one). */
418     unsigned cbgwidthexpn;
419
420     /* The exponent value for the nominal code block group height
421        This quantity is associated with the next lower resolution level
422        (assuming that there is one). */
423     unsigned cbgheightexpn;
424
425     /* The exponent value for the code block width. */
426     uint_fast16_t cblkwidthexpn;
427
428     /* The exponent value for the code block height. */
429     uint_fast16_t cblkheightexpn;
430
431 } jpc_dec_rlvl_t;
432
433 /* Decoder per-tile-component state information. */
434
435 typedef struct {
436
437     /* The x-coordinate of the top-left corner of the tile-component
438        in the coordinate system of the tile-component. */
439     uint_fast32_t xstart;
440
441     /* The y-coordinate of the top-left corner of the tile-component
442        in the coordinate system of the tile-component. */
443     uint_fast32_t ystart;
444
445     /* The x-coordinate of the bottom-right corner of the tile-component
446        in the coordinate system of the tile-component (plus one). */
447     uint_fast32_t xend;
448
449     /* The y-coordinate of the bottom-right corner of the tile-component
450        in the coordinate system of the tile-component (plus one). */
451     uint_fast32_t yend;
452
453     /* The component data for the current tile. */
454     jas_matrix_t *data;
455
456     /* The number of resolution levels. */
457     unsigned numrlvls;
458
459     /* The per resolution level information. */
460     jpc_dec_rlvl_t *rlvls;
461
462     /* The TSFB. */
463     jpc_tsfb_t *tsfb;
464
465 } jpc_dec_tcomp_t;
466
467 /*
468  * Tile states.
469  */
470
471 #define JPC_TILE_INIT    0
472 #define JPC_TILE_ACTIVE  1
473 #define JPC_TILE_ACTIVELAST  2
474 #define JPC_TILE_DONE    3
475
476 /* Decoder per-tile state information. */
477
478 typedef struct {
479
480     /* The processing state for this tile. */
481     int state;
482
483     /* The x-coordinate of the top-left corner of the tile on the reference
484        grid. */
485     uint_fast32_t xstart;
486
487     /* The y-coordinate of the top-left corner of the tile on the reference
488        grid. */
489     uint_fast32_t ystart;
490
491     /* The x-coordinate of the bottom-right corner of the tile on the
492        reference grid (plus one). */
493     uint_fast32_t xend;
494

```

```

495     /* The y-coordinate of the bottom-right corner of the tile on the
496        reference grid (plus one). */
497     uint_fast32_t yend;
498
499     /* The packed packet header data for this tile. */
500     jpc_ppxstab_t *pptstab;
501
502     /* A stream containing the packed packet header data for this tile. */
503     jas_stream_t *pkthdrstream;
504
505     /* The coding parameters for this tile. */
506     jpc_dec_cp_t *cp;
507
508     /* The per tile-component information. */
509     jpc_dec_tcomp_t *tcomps;
510
511     /* The next expected tile-part number. */
512     unsigned partno;
513
514     /* The number of tile-parts. */
515     unsigned numparts;
516
517     /* The coding mode. */
518     int realmode;
519
520     /* The packet iterator for this tile. */
521     jpc_pi_t *pi;
522 } jpc_dec_tile_t;
523
524 /* Decoder per-component state information. */
525
526 typedef struct {
527     /* The horizontal sampling period. */
528     uint_fast32_t hstep;
529
530     /* The vertical sampling period. */
531     uint_fast32_t vstep;
532
533     /* The number of samples in the horizontal direction. */
534     uint_fast32_t width;
535
536     /* The number of samples in the vertical direction. */
537     uint_fast32_t height;
538
539     /* The precision of the sample data. */
540     uint_fast16_t prec;
541
542     /* The signedness of the sample data. */
543     bool sgnd;
544
545     /* The sample alignment horizontal offset. */
546     uint_fast32_t hsubstep;
547
548     /* The sample alignment vertical offset. */
549     uint_fast32_t vsubstep;
550 } jpc_dec_cmpt_t;
551
552 /* Decoder state information. */
553
554 typedef struct {
555     /* The decoded image. */
556     jas_image_t *image;
557
558     /* The x-coordinate of the top-left corner of the image area on
559        the reference grid. */
560     uint_fast32_t xstart;
561
562     /* The y-coordinate of the top-left corner of the image area on
563        the reference grid. */
564     uint_fast32_t ystart;
565
566     /* The x-coordinate of the bottom-right corner of the image area on
567        the reference grid (plus one). */
568     uint_fast32_t xend;
569
570     /* The y-coordinate of the bottom-right corner of the image area on
571        the reference grid (plus one). */
572     uint_fast32_t yend;
573
574     /* The x-coordinate of the bottom-right corner of the image area on
575        the reference grid (plus one). */
576     uint_fast32_t xend;
577
578     /* The y-coordinate of the bottom-right corner of the image area on
579        the reference grid (plus one). */
580     uint_fast32_t yend;
581 } jpc_dec_state_t;

```

```

576     uint_fast32_t yend;
577
578     /* The nominal tile width in units of the image reference grid. */
579     uint_fast32_t tilewidth;
580
581     /* The nominal tile height in units of the image reference grid. */
582     uint_fast32_t tileheight;
583
584     /* The horizontal offset from the origin of the reference grid to the
585        left side of the first tile. */
586     uint_fast32_t tilexoff;
587
588     /* The vertical offset from the origin of the reference grid to the
589        top side of the first tile. */
590     uint_fast32_t tileyoff;
591
592     /* The number of tiles spanning the image area in the vertical
593        direction. */
594     unsigned numhtiles;
595
596     /* The number of tiles spanning the image area in the horizontal
597        direction. */
598     unsigned numvtiles;
599
600     /* The total number of tiles. */
601     unsigned numtiles;
602
603     /* The per-tile information. */
604     jpc_dec_tile_t *tiles;
605
606     /* The tile currently being processed. */
607     jpc_dec_tile_t *curtile;
608
609     /* The number of components. */
610     unsigned numcomps;
611
612     /* The stream containing the input JPEG-2000 code stream data. */
613     jas_stream_t *in;
614
615     /* The default coding parameters for all tiles. */
616     jpc_dec_cp_t *cp;
617
618     /* The maximum number of layers that may be decoded. */
619     unsigned maxlyrs;
620
621     /* The maximum number of packets that may be decoded. */
622     int maxpkts;
623
624     /* The number of packets decoded so far in the processing of the entire
625        code stream. */
626     unsigned numpkts;
627
628     /* The next expected PPM marker segment sequence number. */
629     unsigned ppmseqno;
630
631     /* The current state for code stream processing. */
632     int state;
633
634     /* The per-component information. */
635     jpc_dec_cmpt_t *cmpts;
636
637     /* The information from PPM marker segments. */
638     jpc_ppxstab_t *ppmstab;
639
640     /* A list of streams containing packet header data from PPM marker
641        segments. */
642     jpc_streamlist_t *pkthdrstreams;
643
644     /* The expected ending offset for a tile-part. */
645     long curtileendoff;
646
647     /* This is required by the tier-2 decoder. */
648     jpc_cstate_t *cstate;
649
650     size_t max_samples;
651
652 } jpc_dec_t;
653
654 /* Decoder options. */
655
656 typedef struct {

```

```

657
658     /* The debug level for the decoder. */
659     int debug;
660
661     /* The maximum number of layers to decode. */
662     unsigned maxlyrs;
663
664     /* The maximum number of packets to decode. */
665     int maxpkts;
666
667     size_t max_samples;
668
669 } jpc_dec_importopts_t;
670
671 /*****\
672 * Functions.
673 \*****/
674
675 /* Create a decoder segment object. */
676 jpc_dec_seg_t *jpc_seg_alloc(void);
677
678 /* Destroy a decoder segment object. */
679 void jpc_seg_destroy(jpc_dec_seg_t *seg);
680
681 /* Remove a segment from a segment list. */
682 void jpc_seglist_remove(jpc_dec_seglist_t *list, jpc_dec_seg_t *node);
683
684 /* Insert a segment into a segment list. */
685 void jpc_seglist_insert(jpc_dec_seglist_t *list, jpc_dec_seg_t *ins,
686     jpc_dec_seg_t *node);
687
688 #endif

```

16.51 jpc_enc.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10  * JasPer License Version 2.0
11  *
12  * Copyright (c) 2001-2006 Michael David Adams
13  * Copyright (c) 1999-2000 Image Power, Inc.
14  * Copyright (c) 1999-2000 The University of British Columbia
15  *
16  * All rights reserved.
17  *
18  * Permission is hereby granted, free of charge, to any person (the
19  * "User") obtaining a copy of this software and associated documentation
20  * files (the "Software"), to deal in the Software without restriction,
21  * including without limitation the rights to use, copy, modify, merge,
22  * publish, distribute, and/or sell copies of the Software, and to permit
23  * persons to whom the Software is furnished to do so, subject to the
24  * following conditions:
25  *
26  * 1. The above copyright notices and this permission notice (which
27  * includes the disclaimer below) shall be included in all copies or
28  * substantial portions of the Software.
29  *
30  * 2. The name of a copyright holder shall not be used to endorse or
31  * promote products derived from the Software without specific prior
32  * written permission.
33  *
34  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39  * PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,

```

```

43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 /*
65 * $Id$
66 */
67
68 #ifndef JPC_ENC_H
69 #define JPC_ENC_H
70
71 /*****
72 * Includes.
73 *****/
74
75 #include "jasper/jas_image.h"
76 #include "jasper/jas_seq.h"
77 #include "jasper/jas_stream.h"
78
79 #include "jpc_cod.h"
80 #include "jpc_t1cod.h"
81 #include "jpc_t2cod.h"
82 #include "jpc_mqenc.h"
83 #include "jpc_tagtree.h"
84 #include "jpc_cs.h"
85 #include "jpc_fix.h"
86 #include "jpcflt.h"
87 #include "jpc_tsfb.h"
88
89 /*****
90 * Constants.
91 *****/
92
93 /* The number of bits used in various lookup tables. */
94 #define JPC_NUMEXTRABITS JPC_NMSEDEC_FRACBITS
95
96 /* An invalid R-D slope value. */
97 #define JPC_BADRDSLOPE (-1)
98
99 /*****
100 * Coding parameters types.
101 *****/
102
103 /* Per-component coding paramters. */
104
105 typedef struct {
106
107     /* The horizontal sampling period. */
108     uint_fast8_t sampgrdstepx;
109
110     /* The vertical sampling period. */
111     uint_fast8_t sampgrdstepy;
112
113     /* The sample alignment horizontal offset. */
114     uint_fast8_t sampgrdsubstepx;
115
116     /* The sample alignment vertical offset. */
117     uint_fast8_t sampgrdsubstepy;
118
119     /* The precision of the samples. */
120     uint_fast8_t prec;
121
122     /* The signedness of the samples. */
123     bool sgnd;

```



```

124
125     /* The number of step sizes. */
126     uint_fast16_t numstepsizes;
127
128     /* The quantizer step sizes. */
129     uint_fast16_t stepsizes[JPC_MAXBANDS];
130
131 } jpc_enc_ccp_t;
132
133 /* Per-tile coding parameters. */
134
135 typedef struct {
136
137     /* The coding mode. */
138     bool intmode;
139
140     /* The coding style (i.e., SOP, EPH). */
141     uint_fast8_t csty;
142
143     /* The progression order. */
144     uint_fast8_t prg;
145
146     /* The multicomponent transform. */
147     uint_fast8_t mctid;
148
149     /* The number of layers. */
150     uint_fast16_t numlyrs;
151
152     /* The normalized bit rates associated with the various
153        intermediate layers. */
154     jpc_fix_t *ilyrrates;
155
156 } jpc_enc_tcp_t;
157
158 /* Per tile-component coding parameters. */
159
160 typedef struct {
161
162     /* The coding style (i.e., explicit precinct sizes). */
163     uint_fast8_t csty;
164
165     /* The maximum number of resolution levels allowed. */
166     uint_fast8_t maxrlvls;
167
168     /* The exponent for the nominal code block width. */
169     uint_fast16_t cblkwidthexpn;
170
171     /* The exponent for the nominal code block height. */
172     uint_fast16_t cblkheightexpn;
173
174     /* The code block style parameters (e.g., lazy, terminate all,
175        segmentation symbols, causal, reset probability models). */
176     uint_fast8_t cblksty;
177
178     /* The QMFB. */
179     uint_fast8_t qmfbid;
180
181     /* The precinct width values. */
182     uint_fast16_t prcwidthexpns[JPC_MAXRLVLS];
183
184     /* The precinct height values. */
185     uint_fast16_t prcheightexpns[JPC_MAXRLVLS];
186
187     /* The number of guard bits. */
188     uint_fast8_t numgbits;
189
190 } jpc_enc_tccp_t;
191
192 /* Coding parameters. */
193
194 typedef struct {
195
196     /* The debug level. */
197     int debug;
198
199     /* The horizontal offset from the origin of the reference grid to the
200        left edge of the image area. */
201     uint_fast32_t imgareatlx;
202
203     /* The vertical offset from the origin of the reference grid to the
204        top edge of the image area. */

```

```

205     uint_fast32_t imgareatly;
206
207     /* The horizontal offset from the origin of the reference grid to the
208        right edge of the image area (plus one). */
209     uint_fast32_t refgrdwidth;
210
211     /* The vertical offset from the origin of the reference grid to the
212        bottom edge of the image area (plus one). */
213     uint_fast32_t refgrdheight;
214
215     /* The horizontal offset from the origin of the tile grid to the
216        origin of the reference grid. */
217     uint_fast32_t tilegrdoffx;
218
219     /* The vertical offset from the origin of the tile grid to the
220        origin of the reference grid. */
221     uint_fast32_t tilegrdoffy;
222
223     /* The nominal tile width in units of the image reference grid. */
224     uint_fast32_t tilewidth;
225
226     /* The nominal tile height in units of the image reference grid. */
227     uint_fast32_t tileheight;
228
229     /* The number of tiles spanning the image area in the horizontal
230        direction. */
231     uint_fast32_t numhtiles;
232
233     /* The number of tiles spanning the image area in the vertical
234        direction. */
235     uint_fast32_t numvtiles;
236
237     /* The number of tiles. */
238     uint_fast32_t numtiles;
239
240     /* The number of components. */
241     uint_fast16_t numcmpts;
242
243     /* The per-component coding parameters. */
244     jpc_enc_ccp_t *ccps;
245
246     /* The per-tile coding parameters. */
247     jpc_enc_tcp_t tcp;
248
249     /* The per-tile-component coding parameters. */
250     jpc_enc_tccp_t tccp;
251
252     /* The target code stream length in bytes. */
253     uint_fast32_t totalsize;
254
255     /* The raw (i.e., uncompressed) size of the image in bytes. */
256     uint_fast32_t rawsize;
257 } jpc_enc_cp_t;
258
259 /*****
260  * Encoder class.
261  *****/
262 /* Encoder per-coding-pass state information. */
263
264 typedef struct {
265
266     /* The starting offset for this pass. */
267     int start;
268
269     /* The ending offset for this pass. */
270     int end;
271
272     /* The type of data in this pass (i.e., MQ or raw). */
273     enum jpc_segtype type;
274
275     /* Flag indicating that this pass is terminated. */
276     int term;
277
278     /* The entropy coder state after coding this pass. */
279     jpc_mqencstate_t mqencstate;
280
281     /* The layer to which this pass has been assigned. */
282     unsigned lyrno;
283 }
284

```

```

286     /* The R-D slope for this pass. */
287     jpc_flt_t rdslope;
288
289     /* The weighted MSE reduction associated with this pass. */
290     jpc_flt_t wmsedec;
291
292     /* The cumulative weighted MSE reduction. */
293     jpc_flt_t cumwmsedec;
294
295     /* The normalized MSE reduction. */
296     long nmsedec;
297
298 } jpc_enc_pass_t;
299
300 /* Encoder per-code-block state information. */
301
302 typedef struct {
303
304     /* The number of passes. */
305     unsigned numpasses;
306
307     /* The per-pass information. */
308     jpc_enc_pass_t *passes;
309
310     /* The number of passes encoded so far. */
311     int numencpasses;
312
313     /* The number of insignificant MSBs. */
314     int numimsbs;
315
316     /* The number of bits used to encode pass data lengths. */
317     int numlenbits;
318
319     /* The byte stream for this code block. */
320     jas_stream_t *stream;
321
322     /* The entropy encoder. */
323     jpc_mqenc_t *mqenc;
324
325     /* The data for this code block. */
326     jas_matrix_t *data;
327
328     /* The state for this code block. */
329     jas_matrix_t *flags;
330
331     /* The number of bit planes required for this code block. */
332     int numbps;
333
334     /* The next pass to be encoded. */
335     jpc_enc_pass_t *curpass;
336
337     /* The per-code-block-group state information. */
338     struct jpc_enc_prc_s *prc;
339
340     /* The saved current pass. */
341     /* This is used by the rate control code. */
342     jpc_enc_pass_t *savedcurpass;
343
344     /* The saved length indicator size. */
345     /* This is used by the rate control code. */
346     int savednumlenbits;
347
348     /* The saved number of encoded passes. */
349     /* This is used by the rate control code. */
350     int savednumencpasses;
351
352 } jpc_enc_cblk_t;
353
354 /* Encoder per-code-block-group state information. */
355
356 typedef struct jpc_enc_prc_s {
357
358     /* The x-coordinate of the top-left corner of the precinct. */
359     uint_fast32_t tlx;
360
361     /* The y-coordinate of the top-left corner of the precinct. */
362     uint_fast32_t tly;
363
364     /* The x-coordinate of the bottom-right corner of the precinct
365        (plus one). */
366     uint_fast32_t brx;

```

```

367
368     /* The y-coordinate of the bottom-right corner of the precinct
369        (plus one). */
370     uint_fast32_t bry;
371
372     /* The number of code blocks spanning the precinct in the horizontal
373        direction. */
374     int numhcbcls;
375
376     /* The number of code blocks spanning the precinct in the vertical
377        direction. */
378     int numvcblcls;
379
380     /* The total number of code blocks. */
381     unsigned numcblcls;
382
383     /* The per-code-block information. */
384     jpc_enc_cblk_t *cblcls;
385
386     /* The inclusion tag tree. */
387     jpc_tagtree_t *incltree;
388
389     /* The insignificant MSBs tag tree. */
390     jpc_tagtree_t *nlibtree;
391
392     /* The per-band information. */
393     struct jpc_enc_band_s *band;
394
395     /* The saved inclusion tag tree. */
396     /* This is used by rate control. */
397     jpc_tagtree_t *savincltree;
398
399     /* The saved leading-insignificant-bit-planes tag tree. */
400     /* This is used by rate control. */
401     jpc_tagtree_t *savnlibtree;
402
403 } jpc_enc_prc_t;
404
405 /* Encoder per-band state information. */
406
407 typedef struct jpc_enc_band_s {
408
409     /* The per precinct information. */
410     jpc_enc_prc_t *prcs;
411
412     /* The coefficient data for this band. */
413     jas_matrix_t *data;
414
415     /* The orientation of this band (i.e., LL, LH, HL, or HH). */
416     enum jpc_tsfb_orient orient;
417
418     /* The number of bit planes associated with this band. */
419     int numbps;
420
421     /* The quantizer step size. */
422     jpc_fix_t absstepsize;
423
424     /* The encoded quantizer step size. */
425     int stepsize;
426
427     /* The L2 norm of the synthesis basis functions associated with
428        this band. (The MCT is not considered in this value.) */
429     jpc_fix_t synweight;
430
431     /* The analysis gain for this band. */
432     int analgain;
433
434     /* The per-resolution-level information. */
435     struct jpc_enc_rlvl_s *rlvl;
436
437 } jpc_enc_band_t;
438
439 /* Encoder per-resolution-level state information. */
440
441 typedef struct jpc_enc_rlvl_s {
442
443     /* The x-coordinate of the top-left corner of the tile-component
444        at this resolution. */
445     uint_fast32_t tlx;
446
447     /* The y-coordinate of the top-left corner of the tile-component

```

```

448         at this resolution. */
449     uint_fast32_t tly;
450
451     /* The x-coordinate of the bottom-right corner of the tile-component
452        at this resolution (plus one). */
453     uint_fast32_t brx;
454
455     /* The y-coordinate of the bottom-right corner of the tile-component
456        at this resolution (plus one). */
457     uint_fast32_t bry;
458
459     /* The exponent value for the nominal precinct width measured
460        relative to the associated LL band. */
461     int prcwidthexpn;
462
463     /* The exponent value for the nominal precinct height measured
464        relative to the associated LL band. */
465     int prcheightexpn;
466
467     /* The number of precincts spanning the resolution level in the
468        horizontal direction. */
469     int numhprcs;
470
471     /* The number of precincts spanning the resolution level in the
472        vertical direction. */
473     int numvprcs;
474
475     /* The total number of precincts. */
476     unsigned numprcs;
477
478     /* The exponent value for the nominal code block group width.
479        This quantity is associated with the next lower resolution level
480        (assuming that there is one). */
481     unsigned cbgwidthexpn;
482
483     /* The exponent value for the nominal code block group height.
484        This quantity is associated with the next lower resolution level
485        (assuming that there is one). */
486     unsigned cbgheightexpn;
487
488     /* The exponent value for the code block width. */
489     uint_fast16_t cblkwidthexpn;
490
491     /* The exponent value for the code block height. */
492     uint_fast16_t cblkheightexpn;
493
494     /* The number of bands associated with this resolution level. */
495     unsigned numbands;
496
497     /* The per-band information. */
498     jpc_enc_band_t *bands;
499
500     /* The parent tile-component. */
501     struct jpc_enc_tcmpt_s *tcmt;
502
503 } jpc_enc_rlvl_t;
504
505 /* Encoder per-tile-component state information. */
506 typedef struct jpc_enc_tcmpt_s {
507
508     /* The number of resolution levels. */
509     unsigned numrlvls;
510
511     /* The per-resolution-level information. */
512     jpc_enc_rlvl_t *rlvls;
513
514     /* The tile-component data. */
515     jas_matrix_t *data;
516
517     /* The QMFB. */
518     int qmfbid;
519
520     /* The number of bands. */
521     int numbands;
522
523     /* The TSFB. */
524     jpc_tsfb_t *tsfb;
525
526     /* The synthesis energy weight (for the MCT). */
527     jpc_fix_t synweight;
528

```

```

529
530     /* The precinct width exponents. */
531     int prcwidthexpns[JPC_MAXRLVLS];
532
533     /* The precinct height exponents. */
534     int prcheightexpns[JPC_MAXRLVLS];
535
536     /* The code block width exponent. */
537     int cblkwidthexpn;
538
539     /* The code block height exponent. */
540     int cblkheightexpn;
541
542     /* Coding style (i.e., explicit precinct sizes). */
543     int csty;
544
545     /* Code block style. */
546     int cblksty;
547
548     /* The number of quantizer step sizes. */
549     int numstepsizes;
550
551     /* The encoded quantizer step sizes. */
552     uint_fast16_t stepsizes[JPC_MAXBANDS];
553
554     /* The parent tile. */
555     struct jpc_enc_tile_s *tile;
556
557 } jpc_enc_tcmpt_t;
558
559 /* Encoder per-tile state information. */
560
561 typedef struct jpc_enc_tile_s {
562
563     /* The tile number. */
564     uint_fast32_t tileno;
565
566     /* The x-coordinate of the top-left corner of the tile measured with
567        respect to the reference grid. */
568     uint_fast32_t tlx;
569
570     /* The y-coordinate of the top-left corner of the tile measured with
571        respect to the reference grid. */
572     uint_fast32_t tly;
573
574     /* The x-coordinate of the bottom-right corner of the tile measured
575        with respect to the reference grid (plus one). */
576     uint_fast32_t brx;
577
578     /* The y-coordinate of the bottom-right corner of the tile measured
579        with respect to the reference grid (plus one). */
580     uint_fast32_t bry;
581
582     /* The coding style. */
583     uint_fast8_t csty;
584
585     /* The progression order. */
586     uint_fast8_t prg;
587
588     /* The number of layers. */
589     unsigned numlyrs;
590
591     /* The MCT to employ (if any). */
592     uint_fast8_t mctid;
593
594     /* The packet iterator (used to determine the order of packet
595        generation). */
596     jpc_pi_t *pi;
597
598     /* The coding mode (i.e., integer or real). */
599     bool intmode;
600
601     /* The number of bytes to allocate to the various layers. */
602     uint_fast32_t *lyrsizes;
603
604     /* The number of tile-components. */
605     unsigned numtcmts;
606
607     /* The per tile-component information. */
608     jpc_enc_tcmpt_t *tcmts;
609

```

```

610         /* The raw (i.e., uncompressed) size of this tile. */
611         uint_fast32_t rawsize;
612
613     } jpc_enc_tile_t;
614
615     /* Encoder class. */
616
617     typedef struct jpc_enc_s {
618
619         /* The image being encoded. */
620         jas_image_t *image;
621
622         /* The output stream. */
623         jas_stream_t *out;
624
625         /* The coding parameters. */
626         jpc_enc_cp_t *cp;
627
628         /* The tile currently being processed. */
629         jpc_enc_tile_t *curtile;
630
631         /* The code stream state. */
632         jpc_cstate_t *cstate;
633
634         /* The number of bytes output so far. */
635         uint_fast32_t len;
636
637         /* The number of bytes available for the main body of the code stream. */
638         /* This is used for rate allocation purposes. */
639         uint_fast32_t mainbodysize;
640
641         /* The marker segment currently being processed. */
642         /* This member is a convenience for making cleanup easier. */
643         jpc_ms_t *mrk;
644
645         /* The stream used to temporarily hold tile-part data. */
646         jas_stream_t *tmpstream;
647     } jpc_enc_t;
648
649
650 #endif

```

16.52 jpc_fix.h

```

1  /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8  /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *

```

```

34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 /*
65 * Fixed-Point Number Class
66 *
67 * $Id$
68 */
69
70 #ifndef JPC_FIX_H
71 #define JPC_FIX_H
72
73 /*****
74 * Includes.
75 *****/
76
77 #include "jasper/jas_config.h"
78 #include "jasper/jas_types.h"
79 #include "jasper/jas_fix.h"
80 #include "jasper/jas_math.h"
81
82 /*****
83 * Basic parameters of the fixed-point type.
84 *****/
85
86 /* The integral type used to represent a fixed-point number. This
87 type must be capable of representing values from  $-(2^{31})$  to  $2^{31}-1$ 
88 (inclusive). */
89 #ifdef JAS_ENABLE_32BIT
90 typedef int_least32_t jpc_fix_t;
91 #else
92 typedef int_fast32_t jpc_fix_t;
93 #endif
94
95 /* The integral type used to represent higher-precision intermediate results.
96 This type should be capable of representing values from  $-(2^{63})$  to  $2^{63}-1$ 
97 (inclusive). */
98 typedef int_fast64_t jpc_fix_big_t;
99
100 /* The number of bits used for the fractional part of a fixed-point number. */
101 #define JPC_FIX_FRACBITS 13
102
103 /*****
104 * Instantiations of the generic fixed-point number macros for the
105 * parameters given above. (Too bad C does not support templates, eh?)
106 * The purpose of these macros is self-evident if one examines the
107 * corresponding macros in the jasper/jas_fix.h header file.
108 *****/
109
110 #define JPC_FIX_ZERO JAS_FIX_ZERO(jpc_fix_t, JPC_FIX_FRACBITS)
111 #define JPC_FIX_ONE JAS_FIX_ONE(jpc_fix_t, JPC_FIX_FRACBITS)
112 #define JPC_FIX_HALF JAS_FIX_HALF(jpc_fix_t, JPC_FIX_FRACBITS)
113
114 JAS_ATTRIBUTE_CONST

```



```

115 static inline jpc_fix_t jpc_inttofix(int x)
116 {
117     return JAS_INTTOFIX(jpc_fix_t, JPC_FIX_FRACBITS, x);
118 }
119
120 JAS_ATTRIBUTE_CONST
121 static inline int jpc_fixtoint(jpc_fix_t x)
122 {
123     return JAS_FIXTOINT(jpc_fix_t, JPC_FIX_FRACBITS, x);
124 }
125
126 JAS_ATTRIBUTE_CONST
127 static inline double jpc_fixtodbl(jpc_fix_t x)
128 {
129     return JAS_FIXTODBL(jpc_fix_t, JPC_FIX_FRACBITS, x);
130 }
131
132 JAS_ATTRIBUTE_CONST
133 static inline jpc_fix_t jpc_dbltofix(double x)
134 {
135     return JAS_DBLTOFIX(jpc_fix_t, JPC_FIX_FRACBITS, x);
136 }
137
138 JAS_ATTRIBUTE_CONST
139 static inline jpc_fix_t jpc_fix_add(jpc_fix_t x, jpc_fix_t y)
140 {
141     return JAS_FIX_ADD(jpc_fix_t, JPC_FIX_FRACBITS, x, y);
142 }
143
144 JAS_ATTRIBUTE_CONST
145 static inline jpc_fix_t jpc_fix_sub(jpc_fix_t x, jpc_fix_t y)
146 {
147     return JAS_FIX_SUB(jpc_fix_t, JPC_FIX_FRACBITS, x, y);
148 }
149
150 JAS_ATTRIBUTE_CONST
151 static inline jpc_fix_t jpc_fix_mul(jpc_fix_big_t x, jpc_fix_big_t y)
152 {
153     return JAS_FIX_MUL(jpc_fix_t, JPC_FIX_FRACBITS, jpc_fix_big_t, x, y);
154 }
155
156 JAS_ATTRIBUTE_CONST
157 static inline jpc_fix_big_t jpc_fix_mulbyint(jpc_fix_big_t x, int y)
158 {
159     return JAS_FIX_MUL(jpc_fix_t, JPC_FIX_FRACBITS, jpc_fix_big_t, x, y);
160 }
161
162 JAS_ATTRIBUTE_CONST
163 static inline jpc_fix_t jpc_fix_div(jpc_fix_big_t x, jpc_fix_t y)
164 {
165     return JAS_FIX_DIV(jpc_fix_t, JPC_FIX_FRACBITS, jpc_fix_big_t, x, y);
166 }
167
168 JAS_ATTRIBUTE_CONST
169 static inline jpc_fix_t jpc_fix_neg(jpc_fix_t x)
170 {
171     return JAS_FIX_NEG(jpc_fix_t, JPC_FIX_FRACBITS, x);
172 }
173
174 // #define      jpc_fix_asl(x, n)      JAS_FIX_ASL(jpc_fix_t, JPC_FIX_FRACBITS, x, n)
175 // #define      jpc_fix_asr(x, n)      JAS_FIX_ASR(jpc_fix_t, JPC_FIX_FRACBITS, x, n)
176
177 #ifdef JAS_ENABLE_32BIT
178 #define jpc_fix_asl jas_least32_asl
179 #define jpc_fix_asr jas_least32_asr
180 #else
181 #define jpc_fix_asl jas_fast32_asl
182 #define jpc_fix_asr jas_fast32_asr
183 #endif
184
185 #define jpc_fix_pluseq(x, y)      JAS_FIX_PLUSEQ(jpc_fix_t, JPC_FIX_FRACBITS, x, y)
186 #define jpc_fix_minuseq(x, y)    JAS_FIX_MINUSEQ(jpc_fix_t, JPC_FIX_FRACBITS, x, y)
187 #define jpc_fix_muleq(x, y)      \
188     JAS_FIX_MULEQ(jpc_fix_t, JPC_FIX_FRACBITS, jpc_fix_big_t, x, y)
189
190 JAS_ATTRIBUTE_CONST
191 static inline jpc_fix_t jpc_fix_abs(jpc_fix_t x)
192 {
193     return JAS_FIX_ABS(jpc_fix_t, JPC_FIX_FRACBITS, x);
194 }
195

```

```

196 JAS_ATTRIBUTE_CONST
197 static inline bool jpc_fix_isint(jpc_fix_t x)
198 {
199     return JAS_FIX_ISINT(jpc_fix_t, JPC_FIX_FRACBITS, x);
200 }
201
202 JAS_ATTRIBUTE_CONST
203 static inline int jpc_fix_sgn(jpc_fix_t x)
204 {
205     return JAS_FIX_SGN(jpc_fix_t, JPC_FIX_FRACBITS, x);
206 }
207
208 JAS_ATTRIBUTE_CONST
209 static inline jpc_fix_t jpc_fix_round(jpc_fix_t x)
210 {
211     return JAS_FIX_ROUND(jpc_fix_t, JPC_FIX_FRACBITS, x);
212 }
213
214 JAS_ATTRIBUTE_CONST
215 static inline jpc_fix_t jpc_fix_floor(jpc_fix_t x)
216 {
217     return JAS_FIX_FLOOR(jpc_fix_t, JPC_FIX_FRACBITS, x);
218 }
219
220 /*****
221  * Extra macros for convenience.
222  *****/
223
224 /* Compute the sum of three fixed-point numbers. */
225 JAS_ATTRIBUTE_CONST
226 static inline jpc_fix_t jpc_fix_add3(jpc_fix_t x, jpc_fix_t y, jpc_fix_t z)
227 {
228     return jpc_fix_add(jpc_fix_add(x, y), z);
229 }
230
231 #endif

```

16.53 jpcflt.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * Jasper License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A

```

```

39  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.  IN NO
40  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.  NO ASSURANCES ARE
45  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49  * PROPERTY RIGHTS OR OTHERWISE.  AS A CONDITION TO EXERCISING THE RIGHTS
50  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY.  THE SOFTWARE
52  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58  * RISK ACTIVITIES").  THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60  *
61  * __END_OF_JASPER_LICENSE__
62  */
63
64  /*
65  * Floating-Point Class
66  *
67  * $Id$
68  */
69
70  #ifndef JPC_FLT_H
71  #define JPC_FLT_H
72
73  /* The code ought to be modified so this type is not used at all. */
74  /* Very few places in the code rely on floating-point arithmetic, aside
75  from conversions in printf's. */
76  typedef double jpc_flt_t;
77
78  #endif

```

16.54 jpc_math.h

```

1  /*
2  * Copyright (c) 2001-2002 Michael David Adams.
3  * All rights reserved.
4  */
5
6  /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE.  NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER.  THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS

```

```

35  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
43  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
48  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
50  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58  *
59  * __END_OF_JASPER_LICENSE__
60  */
61
62 #ifndef JPC_MATH_H
63 #define JPC_MATH_H
64
65 /*****
66  * Includes
67  *****/
68
69 #include "jasper/jas_config.h"
70
71 #include "jpc_fix.h"
72
73 /*****
74  * Macros
75  *****/
76
77 /* Compute the floor of the quotient of two integers. */
78 #define JPC_FLOORDIV(x, y)      ((x) / (y))
79
80 /* Compute the ceiling of the quotient of two integers. */
81 #define JPC_CEILDIV(x, y)      (((x) + (y) - 1) / (y))
82
83 /* Compute the floor of (x / 2^y). */
84 #define JPC_FLOORDIVPOW2(x, y) ((x) >> (y))
85
86 /* Compute the ceiling of (x / 2^y). */
87 #define JPC_CEILDIVPOW2(x, y)  (((x) + (1 << (y)) - 1) >> (y))
88
89 /*****
90  * Functions.
91  *****/
92
93 /* Calculate the bit position of the first leading one in a nonnegative
94  integer. */
95 JAS_ATTRIBUTE_CONST
96 int jpc_int_firstone(int x);
97
98 JAS_ATTRIBUTE_CONST
99 int jpc_fix_firstone(jpc_fix_t x);
100
101 /* Calculate the integer quantity floor(log2(x)), where x is a positive
102  integer. */
103 JAS_ATTRIBUTE_CONST
104 unsigned jpc_floorlog2(uint_fast32_t x);
105
106 #endif

```

16.55 jpc_mct.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of

```

```

3  *   British Columbia.
4  *   Copyright (c) 2001-2002 Michael David Adams.
5  *   All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 *   JasPer License Version 2.0
11 *
12 *   Copyright (c) 2001-2006 Michael David Adams
13 *   Copyright (c) 1999-2000 Image Power, Inc.
14 *   Copyright (c) 1999-2000 The University of British Columbia
15 *
16 *   All rights reserved.
17 *
18 *   Permission is hereby granted, free of charge, to any person (the
19 *   "User") obtaining a copy of this software and associated documentation
20 *   files (the "Software"), to deal in the Software without restriction,
21 *   including without limitation the rights to use, copy, modify, merge,
22 *   publish, distribute, and/or sell copies of the Software, and to permit
23 *   persons to whom the Software is furnished to do so, subject to the
24 *   following conditions:
25 *
26 *   1. The above copyright notices and this permission notice (which
27 *   includes the disclaimer below) shall be included in all copies or
28 *   substantial portions of the Software.
29 *
30 *   2. The name of a copyright holder shall not be used to endorse or
31 *   promote products derived from the Software without specific prior
32 *   written permission.
33 *
34 *   THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 *   LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 *   THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 *   "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 *   BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 *   PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 *   EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 *   INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 *   FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 *   NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 *   WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 *   PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 *   THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 *   EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 *   BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 *   PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 *   GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 *   ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 *   IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 *   SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 *   AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 *   SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 *   THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 *   PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 *   RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 *   EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 *   __END_OF_JASPER_LICENSE__
62 */
63
64 /*
65  *   Multicomponent Transform Code
66  *
67  *   $Id$
68  */
69
70 #ifndef JPC_MCT_H
71 #define JPC_MCT_H
72
73 /*****\
74  *   Includes.
75  \*****/
76
77 #include "jpc_fix.h"
78
79 #include "jasper/jas_seq.h"
80
81 /*****\
82  *   Constants.
83  \*****/

```

```

84
85 /*
86  * Multicomponent transform IDs.
87  */
88
89 #define JPC_MCT_NONE    0
90 #define JPC_MCT_ICT     1
91 #define JPC_MCT_RCT     2
92
93 /*****
94  * Functions.
95  *****/
96
97 /* Calculate the forward RCT. */
98 void jpc_rct(jas_matrix_t *c0, jas_matrix_t *c1, jas_matrix_t *c2);
99
100 /* Calculate the inverse RCT. */
101 void jpc_irct(jas_matrix_t *c0, jas_matrix_t *c1, jas_matrix_t *c2);
102
103 /* Calculate the forward ICT. */
104 void jpc_ict(jas_matrix_t *c0, jas_matrix_t *c1, jas_matrix_t *c2);
105
106 /* Calculate the inverse ICT. */
107 void jpc_iict(jas_matrix_t *c0, jas_matrix_t *c1, jas_matrix_t *c2);
108
109 #endif

```

16.56 jpc_mqcod.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2003 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10  * JasPer License Version 2.0
11  *
12  * Copyright (c) 2001-2006 Michael David Adams
13  * Copyright (c) 1999-2000 Image Power, Inc.
14  * Copyright (c) 1999-2000 The University of British Columbia
15  *
16  * All rights reserved.
17  *
18  * Permission is hereby granted, free of charge, to any person (the
19  * "User") obtaining a copy of this software and associated documentation
20  * files (the "Software"), to deal in the Software without restriction,
21  * including without limitation the rights to use, copy, modify, merge,
22  * publish, distribute, and/or sell copies of the Software, and to permit
23  * persons to whom the Software is furnished to do so, subject to the
24  * following conditions:
25  *
26  * 1. The above copyright notices and this permission notice (which
27  * includes the disclaimer below) shall be included in all copies or
28  * substantial portions of the Software.
29  *
30  * 2. The name of a copyright holder shall not be used to endorse or
31  * promote products derived from the Software without specific prior
32  * written permission.
33  *
34  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL

```

```

49  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60  *
61  * __END_OF_JASPER_LICENSE__
62  */
63
64  /*
65  * MQ Arithmetic Coder
66  *
67  * $Id$
68  */
69
70  #ifndef JPC_MQCOD_H
71  #define JPC_MQCOD_H
72
73  /*****\
74  * Includes.
75  \*****/
76
77  #include "jasper/jas_types.h"
78
79  /*****\
80  * Types.
81  \*****/
82
83  /*
84  * MQ coder context information.
85  */
86
87  typedef struct {
88
89      /* The most probable symbol (MPS). */
90      bool mps;
91
92      /* The state index. */
93      int_least8_t ind;
94
95  } jpc_mqctx_t;
96
97  /*
98  * MQ coder state table entry.
99  */
100
101  typedef struct jpc_mqstate_s {
102
103      /* The Qe value. */
104      uint_least16_t qeval;
105
106      /* The MPS. */
107      bool mps;
108
109      /* The NMPS state. */
110      const struct jpc_mqstate_s *nmmps;
111
112      /* The NLPS state. */
113      const struct jpc_mqstate_s *nlps;
114
115  } jpc_mqstate_t;
116
117  /*****\
118  * Data.
119  \*****/
120
121  /* The state table for the MQ coder. */
122  extern const jpc_mqstate_t jpc_mqstates[];
123
124  #endif

```

16.57 jpc_mqdec.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2003 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 /*
65  * MQ Arithmetic Decoder
66  *
67  * $Id$
68  */
69
70 #ifndef JPC_MQDEC_H
71 #define JPC_MQDEC_H
72
73 /*****\
74  * Includes.
75  \*****/
76
77 #include "jasper/jas_types.h"
78 #include "jasper/jas_stream.h"

```



```

79
80 #include "jpc_mqcod.h"
81
82 #include <stdio.h>
83
84 /*****
85  * Types.
86  *****/
87
88 /* MQ arithmetic decoder. */
89
90 typedef struct {
91     /* The C register. */
92     uint_least32_t creg;
93
94     /* The A register. */
95     uint_least32_t areg;
96
97     /* The CT register. */
98     uint_least32_t ctreg;
99
100     /* The current context. */
101     const jpc_mqstate_t **curctx;
102
103     /* The per-context information. */
104     const jpc_mqstate_t **ctxs;
105
106     /* The maximum number of contexts. */
107     unsigned maxctxs;
108
109     /* The stream from which to read data. */
110     jas_stream_t *in;
111
112     /* The last character read. */
113     jas_uchar inbuffer;
114
115     /* The EOF indicator. */
116     bool eof;
117 } jpc_mqdec_t;
118
119 /*****
120  * Functions/macros for construction and destruction.
121  *****/
122
123 /* Create a MQ decoder. */
124 jpc_mqdec_t *jpc_mqdec_create(unsigned maxctxs, jas_stream_t *in);
125
126 /* Destroy a MQ decoder. */
127 void jpc_mqdec_destroy(jpc_mqdec_t *dec);
128
129 /*****
130  * Functions/macros for initialization.
131  *****/
132
133 /* Set the input stream associated with a MQ decoder. */
134 void jpc_mqdec_setinput(jpc_mqdec_t *dec, jas_stream_t *in);
135
136 /* Initialize a MQ decoder. */
137 void jpc_mqdec_init(jpc_mqdec_t *dec);
138
139 /*****
140  * Functions/macros for manipulating contexts.
141  *****/
142
143 /* Set the current context for a MQ decoder. */
144 static inline void jpc_mqdec_setcurctx(jpc_mqdec_t *dec, unsigned ctxno)
145 {
146     dec->curctx = &dec->ctxs[ctxno];
147 }
148
149 /* Set the state information for all contexts of a MQ decoder. */
150 void jpc_mqdec_setctxs(const jpc_mqdec_t *dec, unsigned numctxs, const jpc_mqctx_t *ctxs);
151
152 /*****
153  * Functions/macros for decoding bits.
154  *****/
155
156 /* Decode a symbol. */
157 #ifdef NDEBBUG

```

```

160 #define jpc_mqdec_getbit(dec) \
161     jpc_mqdec_getbit_macro(dec)
162 #else
163 #define jpc_mqdec_getbit(dec) \
164     jpc_mqdec_getbit_func(dec)
165 #endif
166
167 /* Decode a symbol (assuming an unskewed probability distribution). */
168 #ifdef NDEBUG
169 #define jpc_mqdec_getbitnoskew(dec) \
170     jpc_mqdec_getbit_macro(dec)
171 #else
172 #define jpc_mqdec_getbitnoskew(dec) \
173     jpc_mqdec_getbit_func(dec)
174 #endif
175
176 /*****
177  * Functions/macros for debugging.
178  *****/
179
180 /* Dump the MQ decoder state for debugging. */
181 void jpc_mqdec_dump(const jpc_mqdec_t *dec);
182
183 /*****
184  * EVERYTHING BELOW THIS POINT IS IMPLEMENTATION SPECIFIC AND NOT PART OF THE
185  * APPLICATION INTERFACE. DO NOT RELY ON ANY OF THE INTERNAL FUNCTIONS/MACROS
186  * GIVEN BELOW.
187  *****/
188
189 bool jpc_mqdec_mpsexchrenormd(jpc_mqdec_t *dec);
190 bool jpc_mqdec_lpsexchrenormd(jpc_mqdec_t *dec);
191
192 JAS_FORCE_INLINE
193 static bool jpc_mqdec_getbit_macro(jpc_mqdec_t *dec)
194 {
195     const jpc_mqstate_t *const state = *dec->curctx;
196
197     dec->areg -= state->qeval;
198
199     if (dec->creg >= (uint_least32_t)state->qeval << 16) {
200         dec->creg -= (uint_least32_t)state->qeval << 16;
201         return dec->areg & 0x8000
202             ? state->mps
203             : jpc_mqdec_mpsexchrenormd(dec);
204     } else {
205         return jpc_mqdec_lpsexchrenormd(dec);
206     }
207 }
208
209 JAS_FORCE_INLINE
210 static bool jpc_mqdec_mpsexchange(uint_least32_t areg, uint_least32_t delta, const jpc_mqstate_t **curctx)
211 {
212     if (areg < delta) {
213         const jpc_mqstate_t *state = *curctx;
214         /* LPS decoded. */
215         *curctx = state->nmps;
216         return !state->mps;
217     } else {
218         const jpc_mqstate_t *state = *curctx;
219         /* MPS decoded. */
220         *curctx = state->nmps;
221         return state->mps;
222     }
223 }
224
225 JAS_FORCE_INLINE
226 static bool jpc_mqdec_lpsexchange(uint_least32_t *areg_p, uint_least32_t delta, const jpc_mqstate_t
227     **curctx)
228 {
229     if (*areg_p >= delta) {
230         const jpc_mqstate_t *state = *curctx;
231         *areg_p = delta;
232         *curctx = state->nmps;
233         return !state->mps;
234     } else {
235         const jpc_mqstate_t *state = *curctx;
236         *areg_p = delta;
237         *curctx = state->nmps;
238         return state->mps;
239     }
240 }

```

```

240
241 bool jpc_mqdec_getbit_func(jpc_mqdec_t *dec);
242
243 #endif

```

16.58 jpc_mqenc.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 /*
65 * MQ Arithmetic Encoder
66 *
67 * $Id$
68 */
69
70 #ifndef JPC_MQENC_H

```

```

71 #define JPC_MQENC_H
72
73 /*****\
74 * Includes.
75 \*****/
76
77 #include "jasper/jas_types.h"
78 #include "jasper/jas_stream.h"
79
80 #include "jpc_mqcod.h"
81
82 #include <stdio.h>
83
84 /*****\
85 * Constants.
86 \*****/
87
88 /*
89  * Termination modes.
90  */
91
92 #define JPC_MQENC_DEFTERM      0      /* default termination */
93 #define JPC_MQENC_PTERM       1      /* predictable termination */
94
95 /*****\
96 * Types.
97 \*****/
98
99 /* MQ arithmetic encoder class. */
100
101 typedef struct {
102
103     /* The C register. */
104     uint_least32_t creg;
105
106     /* The A register. */
107     uint_least32_t areg;
108
109     /* The CT register. */
110     uint_least32_t ctreg;
111
112     /* The maximum number of contexts. */
113     unsigned maxctxs;
114
115     /* The per-context information. */
116     const jpc_mqstate_t **ctxs;
117
118     /* The current context. */
119     const jpc_mqstate_t **curctx;
120
121     /* The stream for encoder output. */
122     jas_stream_t *out;
123
124     /* The byte buffer (i.e., the B variable in the standard). */
125     int_least16_t outbuf;
126
127     /* The last byte output. */
128     int_least16_t lastbyte;
129
130     /* The error indicator. */
131     bool err;
132
133 } jpc_mqenc_t;
134
135 /* MQ arithmetic encoder state information. */
136
137 typedef struct {
138
139     /* The A register. */
140     unsigned areg;
141
142     /* The C register. */
143     unsigned creg;
144
145     /* The CT register. */
146     unsigned ctreg;
147
148     /* The last byte output by the encoder. */
149     int lastbyte;
150
151 } jpc_mqencstate_t;

```

```

152
153 /*****
154  * Functions/macros for construction and destruction.
155  *****/
156
157 /* Create a MQ encoder. */
158 jpc_mqenc_t *jpc_mqenc_create(unsigned maxctxs, jas_stream_t *out);
159
160 /* Destroy a MQ encoder. */
161 void jpc_mqenc_destroy(jpc_mqenc_t *enc);
162
163 /*****
164  * Functions/macros for initialization.
165  *****/
166
167 /* Initialize a MQ encoder. */
168 void jpc_mqenc_init(jpc_mqenc_t *enc);
169
170 /*****
171  * Functions/macros for context manipulation.
172  *****/
173
174 /* Set the current context. */
175 static inline void jpc_mqenc_setcurctx(jpc_mqenc_t *enc, unsigned ctxno) {
176     enc->curctx = &enc->ctxs[ctxno];
177 }
178
179 /* Set the state information for multiple contexts. */
180 void jpc_mqenc_setctxs(jpc_mqenc_t *enc, unsigned numctxs, const jpc_mqctx_t *ctxs);
181
182 /*****
183  * Miscellaneous functions/macros.
184  *****/
185
186 /* Get the error state of a MQ encoder. */
187 static inline bool jpc_mqenc_error(const jpc_mqenc_t *enc) {
188     return enc->err;
189 }
190
191 /* Get the current encoder state. */
192 void jpc_mqenc_getstate(const jpc_mqenc_t *enc, jpc_mqencstate_t *state);
193
194 /* Terminate the code. */
195 int jpc_mqenc_flush(jpc_mqenc_t *enc, int termmode);
196
197 /*****
198  * Functions/macros for encoding bits.
199  *****/
200
201 /*****
202  * Functions/macros for debugging.
203  *****/
204
205 int jpc_mqenc_dump(const jpc_mqenc_t *mqenc);
206
207 /*****
208  * Implementation-specific details.
209  *****/
210
211 /* Note: These function prototypes are included only to satisfy the
212  needs of the mqenc_putbit_macro macro. Do not call any of these
213  functions directly. */
214 int jpc_mqenc_codemps2(jpc_mqenc_t *enc);
215 int jpc_mqenc_codelps(jpc_mqenc_t *enc);
216
217 int jpc_mqenc_putbit(jpc_mqenc_t *enc, bool bit);
218
219 #endif

```

16.59 jpc_qmfb.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  * British Columbia.
4  * Copyright (c) 2001-2004 Michael David Adams.
5  * All rights reserved.
6  */

```

```

7
8 /* __START_OF_JASPER_LICENSE__
9 *
10 * Jasper License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 /*
65 * Quadrature Mirror-Image Filter Bank (QMFB) Routines
66 *
67 * $Id$
68 */
69
70 #ifndef JPC_QMFB_H
71 #define JPC_QMFB_H
72
73 /*****
74 * Includes.
75 *****/
76
77 #include "jpc_fix.h"
78
79 /*****
80 * Constants.
81 *****/
82
83 /* QMFB IDs. */
84 #define JPC_QMFB1D_FT 1 /* 5/3 */
85 #define JPC_QMFB1D_NS 2 /* 9/7 */
86
87 /*****

```

```

88 * Types.
89 \*****/
90
91 /*****\
92 * Functions.
93 \*****/
94
95 #if !defined(JPC_QMFB_COLGRPSIZE)
96 /* The number of columns to group together during the vertical processing
97 stage of the wavelet transform. */
98 /* The default value for this parameter is probably not optimal for
99 any particular platform. Hopefully, it is not too unreasonable, however. */
100 #define JPC_QMFB_COLGRPSIZE      16
101 #endif
102
103 typedef struct {
104     int (*analyze)(jpc_fix_t *, int, int, int, int, int);
105     int (*synthesize)(jpc_fix_t *, int, int, int, int, int);
106     const double *lpenergywts;
107     const double *hpenergywts;
108 } jpc_qmfb2d_t;
109
110 extern const jpc_qmfb2d_t jpc_ft_qmfb2d;
111 extern const jpc_qmfb2d_t jpc_ns_qmfb2d;
112
113 #endif

```

16.60 jpc_t1cod.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL

```

```

49  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60  *
61  * __END_OF_JASPER_LICENSE__
62  */
63
64  /*
65  * $Id$
66  */
67
68  #ifndef JPC_T1COD_H
69  #define JPC_T1COD_H
70
71  /*****\
72  * Includes.
73  \*****/
74
75  #include "jpc_fix.h"
76  #include "jpc_mqcod.h"
77  #include "jpc_tsfb.h"
78  #include "jasper/jas_math.h"
79
80  /*****\
81  * Constants.
82  \*****/
83
84  /* The number of bits used to index into various lookup tables. */
85  #define JPC_NMSEDEC_BITS 7
86  #define JPC_NMSEDEC_FRACBITS (JPC_NMSEDEC_BITS - 1)
87
88  /*
89  * Segment types.
90  */
91
92  enum jpc_segtype {
93      JPC_SEG_INVALID,
94
95      /* MQ. */
96      JPC_SEG_MQ,
97
98      /* Raw. */
99      JPC_SEG_RAW,
100 };
101
102 /* The nominal word size. */
103 #define JPC_PREC 32
104
105 /* Tier-1 coding pass types. */
106 enum jpc_passtype {
107     JPC_SIGPASS, /*< significance */
108     JPC_REFPASS, /*< refinement */
109     JPC_CLNPASS, /*< cleanup */
110 };
111
112 /*
113  * Per-sample state information for tier-1 coding.
114  */
115
116 /* The northeast neighbour has been found to be significant. */
117 #define JPC_NESIG 0x0001
118 /* The southeast neighbour has been found to be significant. */
119 #define JPC_SESIG 0x0002
120 /* The southwest neighbour has been found to be significant. */
121 #define JPC_SWSIG 0x0004
122 /* The northwest neighbour has been found to be significant. */
123 #define JPC_NWSIG 0x0008
124 /* The north neighbour has been found to be significant. */
125 #define JPC_NSIG 0x0010
126 /* The east neighbour has been found to be significant. */
127 #define JPC_ESIG 0x0020
128 /* The south neighbour has been found to be significant. */
129 #define JPC_SSIG 0x0040

```



```

131 /* The west neighbour has been found to be significant. */
132 #define JPC_WSIG      0x0080
133 /* The significance mask for 8-connected neighbours. */
134 #define JPC_OTHSIGMSK \
135     (JPC_NSIG | JPC_NESIG | JPC_ESIG | JPC_SESIG | JPC_SSIG | JPC_SWSIG | JPC_WSIG | JPC_NWSIG)
136 /* The significance mask for 4-connected neighbours. */
137 #define JPC_PRMSIGMSK (JPC_NSIG | JPC_ESIG | JPC_SSIG | JPC_WSIG)
138
139 /* The north neighbour is negative in value. */
140 #define JPC_NSGN      0x0100
141 /* The east neighbour is negative in value. */
142 #define JPC_ESGN      0x0200
143 /* The south neighbour is negative in value. */
144 #define JPC_SSGN      0x0400
145 /* The west neighbour is negative in value. */
146 #define JPC_WSGN      0x0800
147 /* The sign mask for 4-connected neighbours. */
148 #define JPC_SGNMSK    (JPC_NSGN | JPC_ESGN | JPC_SSGN | JPC_WSGN)
149
150 /* This sample has been found to be significant. */
151 #define JPC_SIG        0x1000
152 /* The sample has been refined. */
153 #define JPC_REFINE     0x2000
154 /* This sample has been processed during the significance pass. */
155 #define JPC_VISIT      0x4000
156
157 /* The number of aggregation contexts. */
158 #define JPC_NUMAGGCTXS 1
159 /* The number of zero coding contexts. */
160 #define JPC_NUMZCCTXS  9
161 /* The number of magnitude contexts. */
162 #define JPC_NUMMAGCTXS 3
163 /* The number of sign coding contexts. */
164 #define JPC_NUMSCCTXS  5
165 /* The number of uniform contexts. */
166 #define JPC_NUMUCTXS   1
167
168 /* The context ID for the first aggregation context. */
169 #define JPC_AGGCTXNO   0
170 /* The context ID for the first zero coding context. */
171 #define JPC_ZCCTXNO    (JPC_AGGCTXNO + JPC_NUMAGGCTXS)
172 /* The context ID for the first magnitude context. */
173 #define JPC_MAGCTXNO   (JPC_ZCCTXNO + JPC_NUMZCCTXS)
174 /* The context ID for the first sign coding context. */
175 #define JPC_SCCTXNO    (JPC_MAGCTXNO + JPC_NUMMAGCTXS)
176 /* The context ID for the first uniform context. */
177 #define JPC_UCTXNO     (JPC_SCCTXNO + JPC_NUMSCCTXS)
178 /* The total number of contexts. */
179 #define JPC_NUMCTXS    (JPC_UCTXNO + JPC_NUMUCTXS)
180
181 /*****
182  * External data.
183  *****/
184
185 /* These lookup tables are used by various macros/functions. */
186 /* Do not access these lookup tables directly. */
187 extern uint_least8_t jpc_zcctxnolut[];
188 extern bool jpc_spblut[];
189 extern uint_least8_t jpc_scctxnolut[];
190 extern uint_least8_t jpc_magctxnolut[];
191 extern jpc_fix_t jpc_refnmsedec[];
192 extern jpc_fix_t jpc_signmsedec[];
193 extern jpc_fix_t jpc_refnmsedec0[];
194 extern jpc_fix_t jpc_signmsedec0[];
195
196 /* The initial settings for the MQ contexts. */
197 extern jpc_mqctx_t jpc_mqctxs[];
198
199 /*****
200  * Functions and macros.
201  *****/
202
203 /* Arithmetic shift right (with ability to shift left also). */
204 JAS_ATTRIBUTE_CONST
205 static inline jpc_fix_t JPC_ASR(jpc_fix_t x, int n)
206 {
207     return n >= 0
208         ? x » n
209         : x « -n;
210 }
211

```

```

212 /* Get the zero coding context. */
213 JAS_ATTRIBUTE_CONST
214 static inline uint_least8_t JPC_GETZCCTXNO(unsigned f, enum jpc_tsfb_orient orient)
215 {
216     return jpc_zcctxnolut[((unsigned)orient << 8) | (f & JPC_OTHSIGMSK)];
217 }
218
219 /* Get the sign prediction bit. */
220 JAS_ATTRIBUTE_CONST
221 static inline bool JPC_GETSPB(unsigned f)
222 {
223     return jpc_spblut[(f & (JPC_PRMSIGMSK | JPC_SGNMSK)) >> 4];
224 }
225
226 /* Get the sign coding context. */
227 JAS_ATTRIBUTE_CONST
228 static inline uint_least8_t JPC_GETSCCTXNO(unsigned f)
229 {
230     return jpc_scctxnolut[(f & (JPC_PRMSIGMSK | JPC_SGNMSK)) >> 4];
231 }
232
233 /* Get the magnitude context. */
234 JAS_ATTRIBUTE_CONST
235 static inline uint_least8_t JPC_GETMAGCTXNO(unsigned f)
236 {
237     return jpc_magctxnolut[(f & JPC_OTHSIGMSK) | (((f & JPC_REFINE) != 0) << 11)];
238 }
239
240 /* Get the normalized MSE reduction for significance passes. */
241 JAS_ATTRIBUTE_CONST
242 static inline jpc_fix_t JPC_GETSIGNMSEDEC(jpc_fix_t x, int bitpos)
243 {
244     return bitpos > JPC_NMSEDEC_FRACBITS
245         ? jpc_signmsecdec[JPC_ASR(x, bitpos - JPC_NMSEDEC_FRACBITS) & JAS_ONES(JPC_NMSEDEC_BITS)]
246         : jpc_signmsecdec0[JPC_ASR(x, bitpos - JPC_NMSEDEC_FRACBITS) & JAS_ONES(JPC_NMSEDEC_BITS)];
247 }
248
249 /* Get the normalized MSE reduction for refinement passes. */
250 JAS_ATTRIBUTE_CONST
251 static inline jpc_fix_t JPC_GETREFNMSEDEC(jpc_fix_t x, int bitpos)
252 {
253     return bitpos > JPC_NMSEDEC_FRACBITS
254         ? jpc_refnmsecdec[JPC_ASR(x, bitpos - JPC_NMSEDEC_FRACBITS) & JAS_ONES(JPC_NMSEDEC_BITS)]
255         : jpc_refnmsecdec0[JPC_ASR(x, bitpos - JPC_NMSEDEC_FRACBITS) & JAS_ONES(JPC_NMSEDEC_BITS)];
256 }
257
258 /* Update the per-sample state information. */
259 static inline void JPC_UPDATEFLAGS4(jpc_fix_t *fp, unsigned rowstep, bool s, bool vcausalflag)
260 {
261     jpc_fix_t *np = fp - rowstep;
262     jpc_fix_t *sp = fp + rowstep;
263     if (vcausalflag) {
264         sp[-1] |= JPC_NESIG;
265         sp[1] |= JPC_NWSIG;
266         if (s) {
267             *sp |= JPC_NSIG | JPC_NSGN;
268             fp[-1] |= JPC_ESIG | JPC_ESGN;
269             fp[1] |= JPC_WSIG | JPC_WSGN;
270         } else {
271             *sp |= JPC_NSIG;
272             fp[-1] |= JPC_ESIG;
273             fp[1] |= JPC_WSIG;
274         }
275     } else {
276         np[-1] |= JPC_SESIG;
277         np[1] |= JPC_SWSIG;
278         sp[-1] |= JPC_NESIG;
279         sp[1] |= JPC_NWSIG;
280         if (s) {
281             *np |= JPC_SSIG | JPC_SSGN;
282             *sp |= JPC_NSIG | JPC_NSGN;
283             fp[-1] |= JPC_ESIG | JPC_ESGN;
284             fp[1] |= JPC_WSIG | JPC_WSGN;
285         } else {
286             *np |= JPC_SSIG;
287             *sp |= JPC_NSIG;
288             fp[-1] |= JPC_ESIG;
289             fp[1] |= JPC_WSIG;
290         }
291     }
292 }

```

```

293
294 /* Initialize the lookup tables used by the codec. */
295 void jpc_initluts(void);
296
297 void jpc_initmqctxs(void);
298
299 /* Get the nominal gain associated with a particular band. */
300 JAS_ATTRIBUTE_CONST
301 unsigned JPC_NOMINALGAIN(unsigned qmfbid, unsigned numlvls, unsigned lvlno, enum jpc_tsfb_orient orient);
302
303 /* Get the coding pass type. */
304 JAS_ATTRIBUTE_CONST
305 enum jpc_passtype JPC_PASSTYPE(unsigned passno);
306
307 /* Get the segment type. */
308 JAS_ATTRIBUTE_CONST
309 enum jpc_segtype JPC_SEGTYPE(unsigned passno, unsigned firstpassno, bool bypass);
310
311 /* Get the number of coding passes in the segment. */
312 JAS_ATTRIBUTE_CONST
313 unsigned JPC_SEGPASSCNT(unsigned passno, unsigned firstpassno, unsigned numpasses, bool bypass,
314     bool termall);
315
316 /* Is the coding pass terminated? */
317 JAS_ATTRIBUTE_CONST
318 bool JPC_ISTERMINATED(unsigned passno, unsigned firstpassno, unsigned numpasses, bool termall,
319     bool lazy);
320
321 #endif

```

16.61 jpc_t1dec.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE

```

```

46  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60  *
61  * __END_OF_JASPER_LICENSE__
62  */
63
64  /*
65  * Tier 1 Decoder
66  *
67  * $Id$
68  */
69
70  #ifndef JPC_T1DEC_H
71  #define JPC_T1DEC_H
72
73  /*****
74  * Includes.
75  *****/
76
77  #include "jpc_dec.h"
78
79  /*****
80  * Functions.
81  *****/
82
83  /* Decode all of the code blocks for a particular tile. */
84  int jpc_dec_decodeblks(jpc_dec_t *dec, jpc_dec_tile_t *tile);
85
86  #endif

```

16.62 jpc_t1enc.h

```

1  /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8  /* __START_OF_JASPER_LICENSE__
9  *
10  * JasPer License Version 2.0
11  *
12  * Copyright (c) 2001-2006 Michael David Adams
13  * Copyright (c) 1999-2000 Image Power, Inc.
14  * Copyright (c) 1999-2000 The University of British Columbia
15  *
16  * All rights reserved.
17  *
18  * Permission is hereby granted, free of charge, to any person (the
19  * "User") obtaining a copy of this software and associated documentation
20  * files (the "Software"), to deal in the Software without restriction,
21  * including without limitation the rights to use, copy, modify, merge,
22  * publish, distribute, and/or sell copies of the Software, and to permit
23  * persons to whom the Software is furnished to do so, subject to the
24  * following conditions:
25  *
26  * 1. The above copyright notices and this permission notice (which
27  * includes the disclaimer below) shall be included in all copies or
28  * substantial portions of the Software.
29  *
30  * 2. The name of a copyright holder shall not be used to endorse or
31  * promote products derived from the Software without specific prior
32  * written permission.
33  *

```

```

34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 /*
65 * Tier 1 Encoder
66 *
67 * $Id$
68 */
69
70 #ifndef JPC_T1ENC_H
71 #define JPC_T1ENC_H
72
73 /*****\
74 * Includes.
75 \*****/
76
77 #include "jpc_enc.h"
78
79 /*****\
80 * Functions.
81 \*****/
82
83 /* Encode all of the code blocks. */
84 int jpc_enc_encblk(jpc_enc_t *enc);
85
86 #endif

```

16.63 jpc_t2cod.h

```

1 /*
2 * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3 * British Columbia.
4 * Copyright (c) 2001-2002 Michael David Adams.
5 * All rights reserved.
6 */
7
8 /* __START_OF_JASPER_LICENSE__
9 *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,

```

```

22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 /*
65 * Tier-2 Coding Library
66 *
67 * $Id$
68 */
69
70 #ifndef JPC_T2COD_H
71 #define JPC_T2COD_H
72
73 /*****\
74 * Includes.
75 \*****/
76
77 #include "jpc_cs.h"
78
79 /*****\
80 * Types.
81 \*****/
82
83 /* Progression change list. */
84
85 typedef struct {
86     /* The number of progression changes. */
87     unsigned numpchgs;
88
89     /* The maximum number of progression changes that can be accomodated
90     without growing the progression change array. */
91     unsigned maxpchgs;
92
93     /* The progression changes. */
94     jpc_pchg_t **pchgs;
95 } jpc_pchglst_t;
96
97 /* Packet iterator per-resolution-level information. */
98
99 typedef struct {
100
101
102

```

```

103     /* The number of precincts. */
104     unsigned numprcs;
105
106     /* The last layer processed for each precinct. */
107     unsigned *prclyrns;
108
109     /* The precinct width exponent. */
110     unsigned prcwidthexpn;
111
112     /* The precinct height exponent. */
113     unsigned prcheightexpn;
114
115     /* The number of precincts spanning the resolution level in the horizontal
116        direction. */
117     unsigned numhprcs;
118 } jpc_pirlvl_t;
119
120
121 /* Packet iterator per-component information. */
122
123 typedef struct {
124
125     /* The number of resolution levels. */
126     unsigned numrlvls;
127
128     /* The per-resolution-level information. */
129     jpc_pirlvl_t *pirlvls;
130
131     /* The horizontal sampling period. */
132     uint_fast32_t hsamp;
133
134     /* The vertical sampling period. */
135     uint_fast32_t vsamp;
136 } jpc_picomp_t;
137
138
139 /* Packet iterator class. */
140
141 typedef struct {
142
143     /* The number of layers. */
144     unsigned numlyrs;
145
146     /* The number of resolution levels. */
147     unsigned maxrlvls;
148
149     /* The number of components. */
150     unsigned numcomps;
151
152     /* The per-component information. */
153     jpc_picomp_t *picomps;
154
155     /* The current component. */
156     jpc_picomp_t *picomp;
157
158     /* The current resolution level. */
159     jpc_pirlvl_t *pirlvl;
160
161     /* The number of the current component. */
162     unsigned compno;
163
164     /* The number of the current resolution level. */
165     unsigned rlvln;
166
167     /* The number of the current precinct. */
168     unsigned prcno;
169
170     /* The number of the current layer. */
171     unsigned lyrno;
172
173     /* The x-coordinate of the current position. */
174     uint_fast32_t x;
175
176     /* The y-coordinate of the current position. */
177     uint_fast32_t y;
178
179     /* The horizontal step size. */
180     uint_fast32_t xstep;
181
182     /* The vertical step size. */
183     uint_fast32_t ystep;

```

```

184
185     /* The x-coordinate of the top-left corner of the tile on the reference
186        grid. */
187     uint_fast32_t xstart;
188
189     /* The y-coordinate of the top-left corner of the tile on the reference
190        grid. */
191     uint_fast32_t ystart;
192
193     /* The x-coordinate of the bottom-right corner of the tile on the
194        reference grid (plus one). */
195     uint_fast32_t xend;
196
197     /* The y-coordinate of the bottom-right corner of the tile on the
198        reference grid (plus one). */
199     uint_fast32_t yend;
200
201     /* The current progression change. */
202     const jpc_pchg_t *pchg;
203
204     /* The progression change list. */
205     jpc_pchglist_t *pchglist;
206
207     /* The progression to use in the absense of explicit specification. */
208     jpc_pchg_t defaultpchg;
209
210     /* The current progression change number. */
211     int pchgno;
212
213     /* Is this the first time in the current progression volume? */
214     bool prgvolfirst;
215
216     /* Is the current iterator value valid? */
217     bool valid;
218
219     /* The current packet number. */
220     int pktno;
221
222 } jpc_pi_t;
223
224 /*****
225  * Functions/macros for packet iterators.
226  *****/
227
228 /* Create a packet iterator. */
229 jpc_pi_t *jpc_pi_create0(void);
230
231 /* Destroy a packet iterator. */
232 void jpc_pi_destroy(jpc_pi_t *pi);
233
234 /* Add a progression change to a packet iterator. */
235 int jpc_pi_addpchg(jpc_pi_t *pi, jpc_pocpchg_t *pchg);
236
237 /* Prepare a packet iterator for iteration. */
238 int jpc_pi_init(jpc_pi_t *pi);
239
240 /* Set the iterator to the first packet. */
241 int jpc_pi_begin(jpc_pi_t *pi);
242
243 /* Proceed to the next packet in sequence. */
244 int jpc_pi_next(jpc_pi_t *pi);
245
246 /* Get the index of the current packet. */
247 #define jpc_pi_getind(pi)      ((pi)->pktno)
248
249 /* Get the component number of the current packet. */
250 #define jpc_pi_cmpno(pi)      (assert(pi->valid), (pi)->compno)
251
252 /* Get the resolution level of the current packet. */
253 #define jpc_pi_rlvlno(pi)      (assert(pi->valid), (pi)->rlvlno)
254
255 /* Get the layer number of the current packet. */
256 #define jpc_pi_lyrno(pi)      (assert(pi->valid), (pi)->lyrno)
257
258 /* Get the precinct number of the current packet. */
259 #define jpc_pi_prcno(pi)      (assert(pi->valid), (pi)->prcno)
260
261 /* Get the progression order for the current packet. */
262 #define jpc_pi_prg(pi)      (assert(pi->valid), (pi)->pchg->prgord)
263
264 /*****

```



```

265 * Functions/macros for progression change lists.
266 \*****/
267
268 /* Create a progression change list. */
269 jpc_pchglst_t *jpc_pchglst_create(void);
270
271 /* Destroy a progression change list. */
272 void jpc_pchglst_destroy(jpc_pchglst_t *pchglst);
273
274 /* Insert a new element into a progression change list. */
275 int jpc_pchglst_insert(jpc_pchglst_t *pchglst, int pchgno, jpc_pchg_t *pchg);
276
277 /* Remove an element from a progression change list. */
278 jpc_pchg_t *jpc_pchglst_remove(jpc_pchglst_t *pchglst, unsigned pchgno);
279
280 /* Get an element from a progression change list. */
281 JAS_ATTRIBUTE_PURE
282 const jpc_pchg_t *jpc_pchglst_get(const jpc_pchglst_t *pchglst, unsigned pchgno);
283
284 /* Copy a progression change list. */
285 jpc_pchglst_t *jpc_pchglst_copy(const jpc_pchglst_t *pchglst);
286
287 /* Get the number of elements in a progression change list. */
288 JAS_ATTRIBUTE_PURE
289 unsigned jpc_pchglst_numpchgs(const jpc_pchglst_t *pchglst);
290
291 \*****/
292 * Functions/macros for progression changes.
293 \*****/
294
295 /* Destroy a progression change. */
296 void jpc_pchg_destroy(jpc_pchg_t *pchg);
297
298 /* Copy a progression change. */
299 jpc_pchg_t *jpc_pchg_copy(const jpc_pchg_t *pchg);
300
301 #endif

```

16.64 jpc_t2dec.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING

```

```

38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 /*
65  * Tier 2 Decoder
66  *
67  * $Id$
68  */
69
70 #ifndef JPC_T2DEC_H
71 #define JPC_T2DEC_H
72
73 /*****\
74  * Includes.
75  \*****/
76
77 #include "jasper/jas_stream.h"
78
79 #include "jpc_dec.h"
80 #include "jpc_t2cod.h"
81
82 /*****\
83  * Functions.
84  \*****/
85
86 /* Decode the packets for a tile-part. */
87 int jpc_dec_decodepkts(jpc_dec_t *dec, jas_stream_t *pkthdrstream,
88     jas_stream_t *in);
89
90 /* Create a packet iterator for the decoder. */
91 jpc_pi_t *jpc_dec_pi_create(jpc_dec_t *dec, jpc_dec_tile_t *tile);
92
93 #endif

```

16.65 jpc_t2enc.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10  * JasPer License Version 2.0
11  *
12  * Copyright (c) 2001-2006 Michael David Adams
13  * Copyright (c) 1999-2000 Image Power, Inc.
14  * Copyright (c) 1999-2000 The University of British Columbia
15  *
16  * All rights reserved.
17  *
18  * Permission is hereby granted, free of charge, to any person (the

```

```

19  * "User") obtaining a copy of this software and associated documentation
20  * files (the "Software"), to deal in the Software without restriction,
21  * including without limitation the rights to use, copy, modify, merge,
22  * publish, distribute, and/or sell copies of the Software, and to permit
23  * persons to whom the Software is furnished to do so, subject to the
24  * following conditions:
25  *
26  * 1. The above copyright notices and this permission notice (which
27  * includes the disclaimer below) shall be included in all copies or
28  * substantial portions of the Software.
29  *
30  * 2. The name of a copyright holder shall not be used to endorse or
31  * promote products derived from the Software without specific prior
32  * written permission.
33  *
34  * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35  * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36  * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37  * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38  * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39  * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40  * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41  * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42  * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43  * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44  * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45  * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46  * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47  * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48  * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49  * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60  *
61  * __END_OF_JASPER_LICENSE__
62  */
63
64 /*
65  * Tier 2 Encoder
66  *
67  * $Id$
68  */
69
70 #ifndef JPC_T2ENC_H
71 #define JPC_T2ENC_H
72
73 /*****\
74  * Includes.
75  \*****/
76
77 #include "jasper/jas_stream.h"
78
79 #include "jpc_enc.h"
80 #include "jpc_t2cod.h"
81
82 /*****\
83  * Functions.
84  \*****/
85
86 /* Encode the packets for a tile. */
87 int jpc_enc_encpkts(jpc_enc_t *enc, jas_stream_t *out);
88
89 /* Encode the specified packet. */
90 int jpc_enc_encpkt(jpc_enc_t *enc, jas_stream_t *out, unsigned compno,
91   unsigned lvlno, unsigned prcno, unsigned lyrno);
92
93 /* Save the tier-2 coding state. */
94 void jpc_save_t2state(jpc_enc_t *enc);
95
96 /* Restore the tier-2 coding state. */
97 void jpc_restore_t2state(jpc_enc_t *enc);
98
99 /* Initialize the tier-2 coding state. */

```

```

100 void jpc_init_t2state(jpc_enc_t *enc, bool raflag);
101
102 /* Create a packet iterator for the encoder. */
103 jpc_pi_t *jpc_enc_pi_create(jpc_enc_cp_t *cp, jpc_enc_tile_t *tile);
104
105 #endif

```

16.66 jpc_tagtree.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60 *
61 * __END_OF_JASPER_LICENSE__
62 */
63
64 /*
65 * Tag Tree Library
66 *
67 * $Id$
68 */

```

```

69
70 #ifndef JPC_TAGTREE_H
71 #define JPC_TAGTREE_H
72
73 /*****\
74 * Includes
75 \*****/
76
77 #include <stdio.h>
78
79 #include "jpc_bs.h"
80
81 /*****\
82 * Constants
83 \*****/
84
85 /* The maximum allowable depth for a tag tree. */
86 #define JPC_TAGTREE_MAXDEPTH 32
87
88 /*****\
89 * Types
90 \*****/
91
92 /*
93  * Tag tree node.
94  */
95
96 typedef struct jpc_tagtreenode_ {
97
98     /* The parent of this node. */
99     struct jpc_tagtreenode_ *parent_;
100
101     /* The value associated with this node. */
102     int value_;
103
104     /* The lower bound on the value associated with this node. */
105     int low_;
106
107     /* A flag indicating if the value is known exactly. */
108     int known_;
109 } jpc_tagtreenode_t;
110
111 /*
112  * Tag tree.
113  */
114
115 typedef struct {
116
117     /* The number of leaves in the horizontal direction. */
118     int numleafsh_;
119
120     /* The number of leaves in the vertical direction. */
121     int numleafsv_;
122
123     /* The total number of nodes in the tree. */
124     int numnodes_;
125
126     /* The nodes. */
127     jpc_tagtreenode_t *nodes_;
128 } jpc_tagtree_t;
129
130
131 /*****\
132 * Functions.
133 \*****/
134
135 /* Create a tag tree. */
136 jpc_tagtree_t *jpc_tagtree_create(int numleafsh, int numleafsv);
137
138 /* Destroy a tag tree. */
139 void jpc_tagtree_destroy(jpc_tagtree_t *tree);
140
141 /* Copy data from one tag tree to another. */
142 void jpc_tagtree_copy(jpc_tagtree_t *dsttree, const jpc_tagtree_t *srctree);
143
144 /* Reset the tag tree state. */
145 void jpc_tagtree_reset(jpc_tagtree_t *tree);
146
147 /* Set the value associated with a particular leaf node of a tag tree. */
148 void jpc_tagtree_setvalue(jpc_tagtree_t *tree, jpc_tagtreenode_t *leaf,

```

```

150     int value);
151
152 /* Get a pointer to a particular leaf node. */
153 JAS_ATTRIBUTE_PURE
154 jpc_tagreenode_t *jpc_tagtree_getleaf(jpc_tagtree_t *tree, int n);
155
156 /* Invoke the tag tree decoding procedure. */
157 int jpc_tagtree_decode(jpc_tagtree_t *tree, jpc_tagreenode_t *leaf,
158     int threshold, jpc_bitstream_t *in);
159
160 /* Invoke the tag tree encoding procedure. */
161 int jpc_tagtree_encode(jpc_tagtree_t *tree, jpc_tagreenode_t *leaf,
162     int threshold, jpc_bitstream_t *out);
163
164 /* Dump a tag tree (for debugging purposes). */
165 void jpc_tagtree_dump(const jpc_tagtree_t *tree, FILE *out);
166
167 #endif

```

16.67 jpc_tsfb.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  *   British Columbia.
4  * Copyright (c) 2001-2004 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HERUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
52 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,

```

```

57  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60  *
61  * __END_OF_JASPER_LICENSE__
62  */
63
64  /*
65  * Tree-Structured Filter Bank (TSFB) Library
66  *
67  * $Id$
68  */
69
70  #ifndef JPC_TSFB_H
71  #define JPC_TSFB_H
72
73  /*****\
74  * Includes.
75  \*****/
76
77  #include "jasper/jas_seq.h"
78  #include "jasper/jas_types.h"
79
80  #include "jpc_fix.h"
81  #include "jpc_qmfb.h"
82
83  /*****\
84  * Constants.
85  \*****/
86
87  #define JPC_TSFB_MAXBANDS      (JPC_TSFB_MAXDEPTH * 3 + 1)
88  #define JPC_TSFB_MAXDEPTH     32
89  #define JPC_TSFB_RITIMODE     JPC_QMFB1D_RITIMODE
90
91  enum jpc_tsfb_orient {
92      JPC_TSFB_LL = 0,
93      JPC_TSFB_LH = 1,
94      JPC_TSFB_HL = 2,
95      JPC_TSFB_HH = 3,
96  };
97
98  /*****\
99  * Types.
100  \*****/
101
102  typedef struct {
103      int xstart;
104      int ystart;
105      int xend;
106      int yend;
107      enum jpc_tsfb_orient orient;
108      int locxstart;
109      int locystart;
110      int locxend;
111      int locyend;
112      jpc_fix_t synenergywt;
113  } jpc_tsfb_band_t;
114
115  typedef struct {
116      unsigned numlvl;
117      const jpc_qmfb2d_t *qmfb;
118  } jpc_tsfb_t;
119
120  /*****\
121  * Functions.
122  \*****/
123
124  /* Create a TSFB. */
125  jpc_tsfb_t *jpc_cod_gettsfb(unsigned qmfbid, unsigned numlevels);
126
127  /* Destroy a TSFB. */
128  void jpc_tsfb_destroy(jpc_tsfb_t *tsfb);
129
130  /* Perform analysis. */
131  int jpc_tsfb_analyze(jpc_tsfb_t *tsfb, jas_seq2d_t *x);
132
133  /* Perform synthesis. */
134  int jpc_tsfb_synthesize(jpc_tsfb_t *tsfb, jas_seq2d_t *x);
135
136  /* Get band information for a TSFB. */
137  int jpc_tsfb_getbands(jpc_tsfb_t *tsfb, uint_fast32_t xstart,

```

```

138     uint_fast32_t ystart, uint_fast32_t xend, uint_fast32_t yend,
139     jpc_tsfb_band_t *bands);
140
141 #endif

```

16.68 jpc_util.h

```

1 /*
2  * Copyright (c) 2001-2002 Michael David Adams.
3  * All rights reserved.
4  */
5
6 /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
43 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
48 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
50 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58 *
59 * __END_OF_JASPER_LICENSE__
60 */
61
62 #ifndef JPC_UTIL_H
63 #define JPC_UTIL_H
64
65 #include "jpc_fix.h"
66
67 #include "jasper/jas_seq.h"
68
69 /* Parse a comma separated list of real numbers into an array of doubles. */
70 int jpc_atoaf(const char *s, int *numvalues, double **values);

```



```

71
72 /* Upsample a sequence. */
73 jas_seq_t *jpc_seq_upsample(const jas_seq_t *seq, int n);
74
75 /* Convolve two sequences. */
76 jas_seq_t *jpc_seq_conv(const jas_seq_t *seq0, const jas_seq_t *seq1);
77
78 /* Compute the norm of a sequence. */
79 JAS_ATTRIBUTE_PURE
80 jpc_fix_t jpc_seq_norm(const jas_seq_t *x);
81
82 #endif

```

16.69 jpg_cod.h

```

1 /*
2  * Copyright (c) 2001-2002 Michael David Adams.
3  * All rights reserved.
4  */
5
6 /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
43 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
48 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
50 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58 *
59 * __END_OF_JASPER_LICENSE__
60 */
61
62 /*

```

```

63  * JPG Format Library
64  *
65  * $Id$
66  */
67
68 #ifndef JPG_COD_H
69 #define JPG_COD_H
70
71 /*****\
72 * Includes.
73 \*****/
74
75 /*****\
76 * Constants.
77 \*****/
78
79 #define JPG_MAGIC      0xffd8
80 #define JPG_MAGICLEN 2
81
82 #endif

```

16.70 jpg_enc.h

```

1  /*
2  * Copyright (c) 2002 Michael David Adams.
3  * All rights reserved.
4  */
5
6  /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
43 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
48 * GRANTED HERUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
50 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,

```

```

55  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58  *
59  * __END_OF_JASPER_LICENSE__
60  */
61
62 #ifndef JPG_ENC_H
63 #define JPG_ENC_H
64
65 typedef struct {
66     int numcmpts;
67     int cmpts[4];
68 } jpg_enc_t;
69
70 #endif

```

16.71 jpg_jpeglib.h

```

1  /*
2  * Copyright (c) 2001-2002 Michael David Adams.
3  * All rights reserved.
4  */
5
6  /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
43 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
48 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
50 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58  */

```

```

59  * __END_OF_JASPER_LICENSE__
60  */
61
62  #ifndef JPG_JPEGLIB_H
63  #define JPG_JPEGLIB_H
64
65  /*****
66  * Includes.
67  *****/
68
69  #include <stdio.h>
70  #include "jasper/jas_types.h"
71
72  /* Note: The jpeglib.h header file does not include definitions of
73     FILE, size_t, etc. */
74  #include <jpeglib.h> /* IWYU pragma: export */
75
76  #endif

```

16.72 mif_cod.h

```

1  /*
2  * Copyright (c) 2001-2002 Michael David Adams.
3  * All rights reserved.
4  */
5
6  /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
43 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
48 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
50 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY

```

```

57  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58  *
59  * __END_OF_JASPER_LICENSE__
60  */
61
62 #ifndef MIF_COD_H
63 #define MIF_COD_H
64
65 /*****\
66  * Includes.
67  \*****/
68
69 #include "jasper/jas_types.h"
70
71 /*****\
72  * Constants.
73  \*****/
74
75 #define MIF_MAGIC                0x4d49460a
76 /* signature */
77
78 #define MIF_MAGICLEN    4
79 /* length of signature in bytes */
80
81 /*****\
82  * Types.
83  \*****/
84
85 /* Per-component information. */
86
87 typedef struct {
88
89     int_fast32_t tlx;
90
91     int_fast32_t tly;
92
93     int_fast32_t width;
94
95     int_fast32_t height;
96
97     int_fast32_t sampperx;
98
99     int_fast32_t samppery;
100
101     int_fast16_t prec;
102
103     int_fast16_t sgnd;
104
105     char *data;
106 } mif_cmpt_t;
107
108
109 /* MIF header. */
110
111 typedef struct {
112
113     uint_fast32_t magic;
114
115     int numcmpts;
116
117     int maxcmpts;
118
119     mif_cmpt_t **cmpts;
120
121 } mif_hdr_t;
122
123 #endif

```

16.73 pgx_cod.h

```

1  /*
2  * Copyright (c) 2001-2002 Michael David Adams.
3  * All rights reserved.
4  */
5
6  /* __START_OF_JASPER_LICENSE__
7  *

```

```

8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
43 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
48 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
50 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58 *
59 * __END_OF_JASPER_LICENSE__
60 */
61
62 /*
63 * PGX Format Library
64 *
65 * $Id$
66 */
67
68 #ifndef PGX_COD_H
69 #define PGX_COD_H
70
71 /*****\
72 * Includes.
73 \*****/
74
75 #include <stdio.h>
76
77 #include "jasper/jas_types.h"
78
79 /*****\
80 * Constants.
81 \*****/
82
83 #define PGX_MAGIC      0x5047
84 #define PGX_MAGICLEN   2
85
86 /*****\
87 * Types.
88 \*****/

```

```

89
90 typedef struct {
91
92     uint_fast16_t magic;
93     /* The signature. */
94
95     bool bigendian;
96     /* The byte ordering used. */
97
98     bool sgnd;
99     /* The signedness of the samples. */
100
101     uint_fast32_t prec;
102     /* The precision of the samples. */
103
104     uint_fast32_t width;
105     /* The width of the component. */
106
107     uint_fast32_t height;
108     /* The height of the component. */
109
110 } pgx_hdr_t;
111
112 /*****\
113 * Functions.
114 \*****/
115
116 void pgx_dumphdr(FILE *out, pgx_hdr_t *hdr);
117
118 #endif

```

16.74 pgx_enc.h

```

1 /*
2  * Copyright (c) 2002 Michael David Adams.
3  * All rights reserved.
4  */
5
6 /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
43 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.

```

```

45 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
48 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
50 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58 *
59 * __END_OF_JASPER_LICENSE__
60 */
61
62 #ifndef PGX_ENC_H
63 #define PGX_ENC_H
64
65 typedef struct {
66     int cmpt;
67 } pgx_enc_t;
68
69 #endif

```

16.75 pnm_cod.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  * British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS

```



```

50  * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51  * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY.  THE SOFTWARE
52  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58  * RISK ACTIVITIES").  THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60  *
61  * __END_OF_JASPER_LICENSE__
62  */
63
64  /*
65  * Portable Pixmap/Graymap Format Support
66  *
67  * $Id$
68  */
69
70  #ifndef PNM_COD_H
71  #define PNM_COD_H
72
73  /*****\
74  * Includes.
75  \*****/
76
77  #include "jasper/jas_types.h"
78
79  /*****\
80  * Constants.
81  \*****/
82
83  /* Magic numbers. */
84  #define PNM_MAGIC_TXTPBM      0x5031 /* Text Portable BitMap (P1) */
85  #define PNM_MAGIC_TXTPGM      0x5032 /* Text Portable GrayMap (P2) */
86  #define PNM_MAGIC_TXTPPM      0x5033 /* Text Portable PixMap (P3) */
87  #define PNM_MAGIC_BINPBM      0x5034 /* Binary Portable BitMap (P4) */
88  #define PNM_MAGIC_BINPGM      0x5035 /* Binary Portable GrayMap (P5) */
89  #define PNM_MAGIC_BINPPM      0x5036 /* Binary Portable PixMap (P6) */
90  #define PNM_MAGIC_PAM         0x5037 /* PAM (P7) */
91
92  /* Type of image data. */
93  #define PNM_TYPE_INVALID      0
94  #define PNM_TYPE_PBM          1 /* BitMap */
95  #define PNM_TYPE_PGM          2 /* GrayMap */
96  #define PNM_TYPE_PPM          3 /* PixMap */
97
98  /* Format of image data. */
99  #define PNM_FMT_TXT           0 /* Text */
100 #define PNM_FMT_BIN           1 /* Binary */
101
102 #define PNM_MAXLINELEN       79
103
104 #define PNM_TUPLETYPE_UNKNOWN  0
105 #define PNM_TUPLETYPE_MONO     1
106 #define PNM_TUPLETYPE_GRAY     2
107 #define PNM_TUPLETYPE_GRAYA    3
108 #define PNM_TUPLETYPE_RGB      4
109 #define PNM_TUPLETYPE_RGBA     5
110
111 /*****\
112 * Types.
113 \*****/
114
115 /* File header. */
116
117 typedef struct {
118     int magic;
119     /* The magic number. */
120
121     int width;
122     /* The image width. */
123
124     int height;
125     /* The image height. */
126
127     int numcmpts;
128
129     int maxval;

```

```

131         /* The maximum allowable sample value. */
132
133     #if 0
134         int tupletype;
135     #endif
136
137     bool sgnd;
138     /* The sample data is signed. */
139
140 } pnm_hdr_t;
141
142 /*****
143  * Functions.
144  *****/
145
146 int pnm_type(uint_fast16_t magic);
147 /* Determine type (i.e., PGM or PPM) from magic number. */
148
149 int pnm_fmt(uint_fast16_t magic);
150 /* Determine format (i.e., text or binary) from magic number. */
151
152 int pnm_maxvaltodepth(uint_fast32_t maxval);
153 /* Determine depth (i.e., precision) from maximum value. */
154
155 #define PNM_ONES(n) \
156     ((n) < 32) ? ((1UL « (n)) - 1) : 0xffffffffUL
157 #endif

```

16.76 pnm_enc.h

```

1  /*
2  * Copyright (c) 2002 Michael David Adams.
3  * All rights reserved.
4  */
5
6  /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE
43 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS

```

```

48 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY.  THE SOFTWARE
50 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56 * RISK ACTIVITIES").  THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58 *
59 * __END_OF_JASPER_LICENSE__
60 */
61
62 #ifndef PNM_ENC_H
63 #define PNM_ENC_H
64
65 typedef struct {
66     int numcmpts;
67     int cmpts[4];
68 } pnm_enc_t;
69
70 #endif

```

16.77 ras_cod.h

```

1 /*
2  * Copyright (c) 1999-2000 Image Power, Inc. and the University of
3  * British Columbia.
4  * Copyright (c) 2001-2002 Michael David Adams.
5  * All rights reserved.
6  */
7
8 /* __START_OF_JASPER_LICENSE__
9  *
10 * JasPer License Version 2.0
11 *
12 * Copyright (c) 2001-2006 Michael David Adams
13 * Copyright (c) 1999-2000 Image Power, Inc.
14 * Copyright (c) 1999-2000 The University of British Columbia
15 *
16 * All rights reserved.
17 *
18 * Permission is hereby granted, free of charge, to any person (the
19 * "User") obtaining a copy of this software and associated documentation
20 * files (the "Software"), to deal in the Software without restriction,
21 * including without limitation the rights to use, copy, modify, merge,
22 * publish, distribute, and/or sell copies of the Software, and to permit
23 * persons to whom the Software is furnished to do so, subject to the
24 * following conditions:
25 *
26 * 1. The above copyright notices and this permission notice (which
27 * includes the disclaimer below) shall be included in all copies or
28 * substantial portions of the Software.
29 *
30 * 2. The name of a copyright holder shall not be used to endorse or
31 * promote products derived from the Software without specific prior
32 * written permission.
33 *
34 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
35 * LICENSE.  NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
36 * THIS DISCLAIMER.  THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
37 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
38 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
39 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS.  IN NO
40 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
41 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
42 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
43 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
44 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.  NO ASSURANCES ARE
45 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
46 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
47 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
48 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
49 * PROPERTY RIGHTS OR OTHERWISE.  AS A CONDITION TO EXERCISING THE RIGHTS
50 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
51 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY.  THE SOFTWARE

```

```

52  * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
53  * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
54  * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
55  * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
56  * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
57  * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
58  * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
59  * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
60  *
61  * __END_OF_JASPER_LICENSE__
62  */
63
64  /*
65  * Sun Rasterfile Library
66  *
67  * $Id$
68  */
69
70  /*****
71  * Sun Rasterfile
72  *****/
73
74  #ifndef RAS_COD_H
75  #define RAS_COD_H
76
77  /*****
78  * Includes.
79  *****/
80
81  #include "jasper/jas_types.h"
82
83  /*****
84  * Constants.
85  *****/
86
87  #define RAS_MAGIC      0x59a66a95
88  #define RAS_MAGICLEN   4
89
90  #define RAS_TYPE_OLD    0
91  #define RAS_TYPE_STD    1
92  #define RAS_TYPE_RLE    2
93
94  #define RAS_MT_NONE     0
95  #define RAS_MT_EQUALRGB 1
96  #define RAS_MT_RAW      2
97
98  /*****
99  * Types.
100  *****/
101
102  /* Sun Rasterfile header. */
103  typedef struct {
104
105      int_fast32_t magic;
106      /* signature */
107
108      int_fast32_t width;
109      /* width of image (in pixels) */
110
111      int_fast32_t height;
112      /* height of image (in pixels) */
113
114      int_fast32_t depth;
115      /* number of bits per pixel */
116
117      int_fast32_t length;
118      /* length of image data (in bytes) */
119
120      int_fast32_t type;
121      /* format of image data */
122
123      int_fast32_t maptype;
124      /* colormap type */
125
126      int_fast32_t maplength;
127      /* length of colormap data (in bytes) */
128  } ras_hdr_t;
129
130
131  #define RAS_CMAP_MAXSIZE 256
132

```

```

133 /* Color map. */
134 typedef struct {
135
136     int len;
137     /* The number of entries in the color map. */
138
139     int data[RAS_CMAP_MAXSIZE];
140     /* The color map data. */
141
142 } ras_cmap_t;
143
144 /*****\
145 * Macros.
146 \*****/
147
148 #define RAS_GETBLUE(x)  (((x) >> 16) & 0xff)
149 #define RAS_GETGREEN(x) (((x) >> 8) & 0xff)
150 #define RAS_GETRED(x)  ((x) & 0xff)
151
152 #define RAS_BLUE(x)     (((x) & 0xff) << 16)
153 #define RAS_GREEN(x)   (((x) & 0xff) << 8)
154 #define RAS_RED(x)     ((x) & 0xff)
155
156 #define RAS_ROWSIZE(hdr) \
157     (((hdr)->width * (hdr)->depth + 15) / 16) * 2)
158 #define RAS_ISRGB(hdr) \
159     ((hdr)->depth == 24 || (hdr)->depth == 32)
160
161 #define RAS_ONES(n) \
162     (((n) == 32) ? 0xffffffffUL : ((1UL << (n)) - 1))
163
164 #endif

```

16.78 ras_enc.h

```

1 /*
2  * Copyright (c) 2002 Michael David Adams.
3  * All rights reserved.
4  */
5
6 /* __START_OF_JASPER_LICENSE__
7  *
8  * JasPer License Version 2.0
9  *
10 * Copyright (c) 2001-2006 Michael David Adams
11 * Copyright (c) 1999-2000 Image Power, Inc.
12 * Copyright (c) 1999-2000 The University of British Columbia
13 *
14 * All rights reserved.
15 *
16 * Permission is hereby granted, free of charge, to any person (the
17 * "User") obtaining a copy of this software and associated documentation
18 * files (the "Software"), to deal in the Software without restriction,
19 * including without limitation the rights to use, copy, modify, merge,
20 * publish, distribute, and/or sell copies of the Software, and to permit
21 * persons to whom the Software is furnished to do so, subject to the
22 * following conditions:
23 *
24 * 1. The above copyright notices and this permission notice (which
25 * includes the disclaimer below) shall be included in all copies or
26 * substantial portions of the Software.
27 *
28 * 2. The name of a copyright holder shall not be used to endorse or
29 * promote products derived from the Software without specific prior
30 * written permission.
31 *
32 * THIS DISCLAIMER OF WARRANTY CONSTITUTES AN ESSENTIAL PART OF THIS
33 * LICENSE. NO USE OF THE SOFTWARE IS AUTHORIZED HEREUNDER EXCEPT UNDER
34 * THIS DISCLAIMER. THE SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS
35 * "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING
36 * BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
37 * PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO
38 * EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
39 * INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
40 * FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
41 * NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
42 * WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. NO ASSURANCES ARE

```

```
43 * PROVIDED BY THE COPYRIGHT HOLDERS THAT THE SOFTWARE DOES NOT INFRINGE
44 * THE PATENT OR OTHER INTELLECTUAL PROPERTY RIGHTS OF ANY OTHER ENTITY.
45 * EACH COPYRIGHT HOLDER DISCLAIMS ANY LIABILITY TO THE USER FOR CLAIMS
46 * BROUGHT BY ANY OTHER ENTITY BASED ON INFRINGEMENT OF INTELLECTUAL
47 * PROPERTY RIGHTS OR OTHERWISE. AS A CONDITION TO EXERCISING THE RIGHTS
48 * GRANTED HEREUNDER, EACH USER HEREBY ASSUMES SOLE RESPONSIBILITY TO SECURE
49 * ANY OTHER INTELLECTUAL PROPERTY RIGHTS NEEDED, IF ANY. THE SOFTWARE
50 * IS NOT FAULT-TOLERANT AND IS NOT INTENDED FOR USE IN MISSION-CRITICAL
51 * SYSTEMS, SUCH AS THOSE USED IN THE OPERATION OF NUCLEAR FACILITIES,
52 * AIRCRAFT NAVIGATION OR COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL
53 * SYSTEMS, DIRECT LIFE SUPPORT MACHINES, OR WEAPONS SYSTEMS, IN WHICH
54 * THE FAILURE OF THE SOFTWARE OR SYSTEM COULD LEAD DIRECTLY TO DEATH,
55 * PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE ("HIGH
56 * RISK ACTIVITIES"). THE COPYRIGHT HOLDERS SPECIFICALLY DISCLAIM ANY
57 * EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.
58 *
59 * __END_OF_JASPER_LICENSE__
60 */
61
62 #ifndef RAS_ENC_H
63 #define RAS_ENC_H
64
65 typedef struct {
66     int numcmpts;
67     int cmpts[4];
68 } ras_enc_t;
69
70 #endif
```