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_copy()`
  - `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_DTYPETOBPC, JP2_BPCTODTYPE
      JAS_IMAGE_CDT_{SETSGND,GETSGND,SETPREC,GETPREC}
      jas_image_cmptdtype
      macros from here
      jas_matrix_setv, jas_matrix_getvref
      jas_matrix_bind{row,col}
      the jpc_fix_ family
      the JPC_QCX and JPC_COX families
```

Chapter 2

License

2.1 License

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

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 sofware, 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% ^
      -DCMAKE_INSTALL_PREFIX=%INSTALL_DIR% %OPTIONS%

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

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

    msbuild %build_dir%\INSTALL.vcxproj

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.

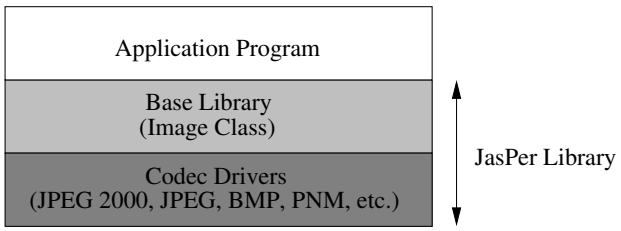

Figure 5.1 Software structure

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_cmprof_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_cmxform_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 provides 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.

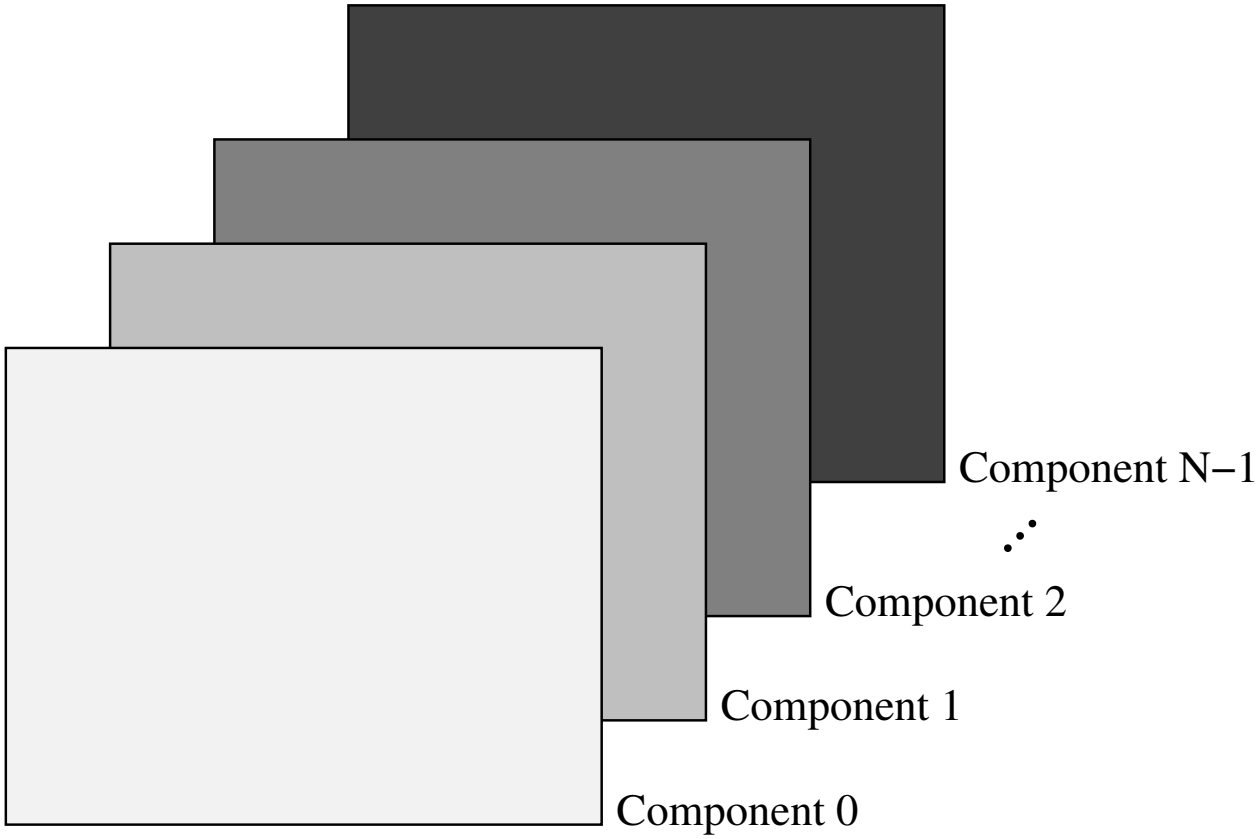

Figure 5.2 Image model: An image with N components.

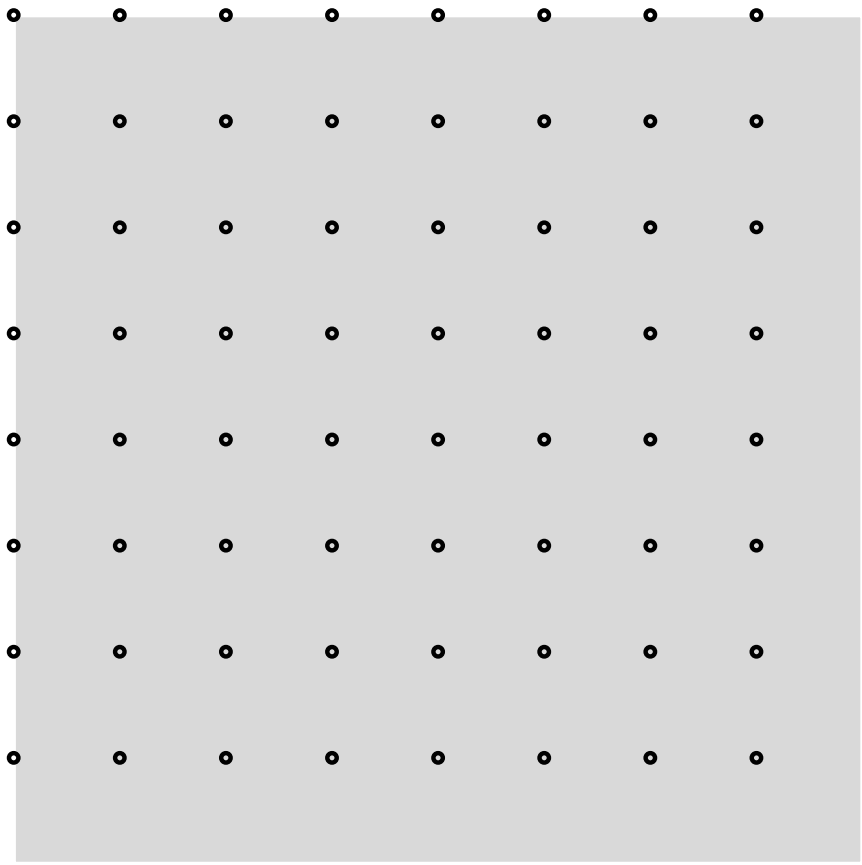

Figure 5.3 Image model: An individual component.

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 kth 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 nothing 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.

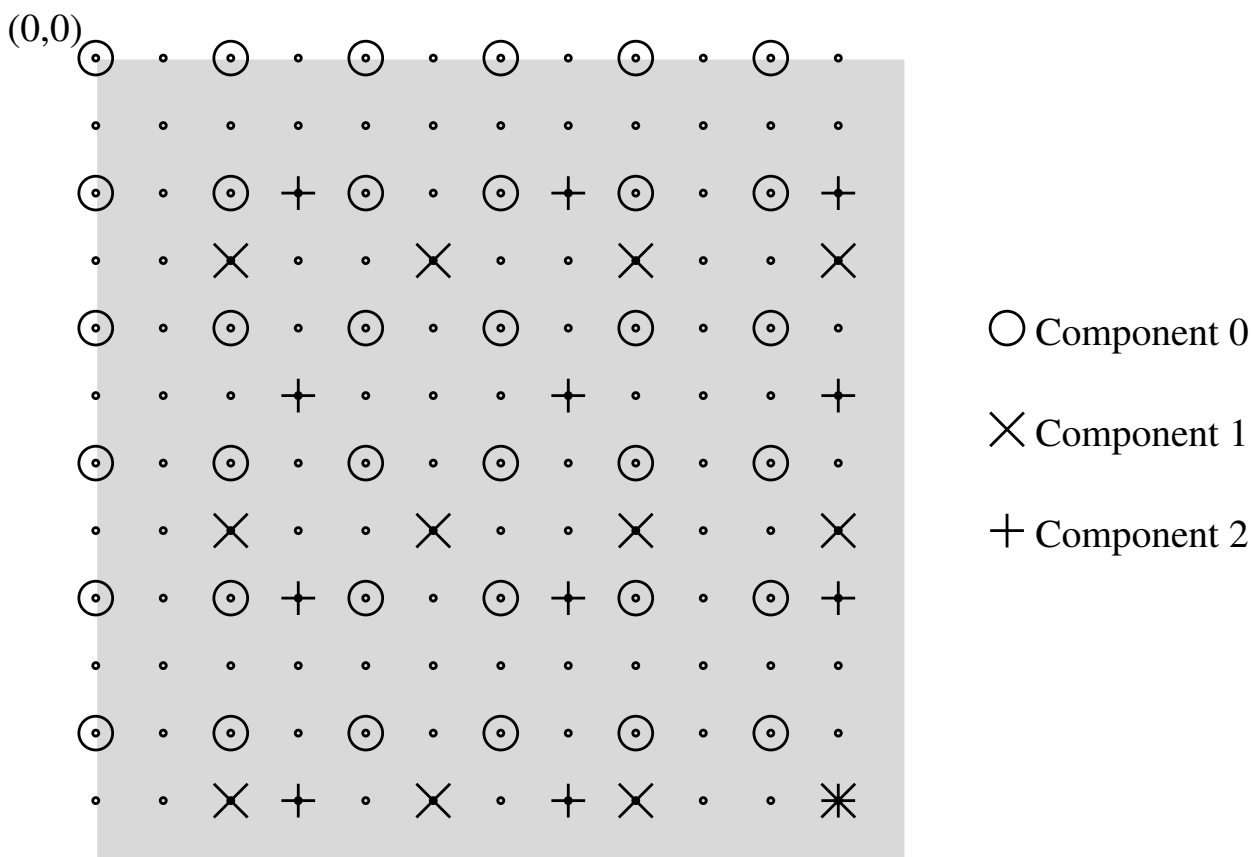

Figure 5.4 Alignment of components on the reference grid.

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 alloator 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 function 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_cmpthstep()
- 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_fmttostr()
- 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_setrwcount()
- jas_stream_getrwlimit()
- jas_stream_getrwcount()

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 $\times$ 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, ..., 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:
  - `lrcp`. 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
  - `cprl`. 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.ijg.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_setrwcount (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:

Chapter 12

Class Index

12.1 Class List

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

Chapter 13

File Index

13.1 File List

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

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

max_mem	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

allocator	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_getrwlimit(stream) (((const jas_stream_t *)(stream))->rwlimit_)

  Get the read/write limit for a stream.

- #define jas_stream_getrwcount(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_setrwlimit (jas_stream_t *stream, long rwlimit)

  Set the read/write limit for a stream.
- JAS_EXPORT long jas_stream_setrwcount (jas_stream_t *stream, long rw_count)

  Set the read/write count for a stream.
- JAS_EXPORT size_t jas_stream_read (jas_stream_t *stream, void *buffer, size_t count)

  Read characters from a stream into a buffer.
- JAS_EXPORT unsigned jas_stream_peek (jas_stream_t *stream, void *buffer, size_t count)

  Attempt to retrieve one or more pending characters of input from a stream into a buffer without actually removing the characters from the stream.
- JAS_EXPORT size_t jas_stream_write (jas_stream_t *stream, const void *buffer, size_t count)

  Write characters from a buffer to a stream.
- JAS_EXPORT int jas_stream_printf (jas_stream_t *stream, const char *format,...)

  Write formatted output to a stream.
- JAS_EXPORT int jas_stream_puts (jas_stream_t *stream, const char *s)

  Write a string to a stream.
- JAS_EXPORT char * jas_stream_gets (jas_stream_t *stream, char *buffer, int buffer_size)

  Read a line of input from a stream.
- JAS_EXPORT int jas_stream_ungetc (jas_stream_t *stream, int c)

  Put a character back on a stream.
- JAS_EXPORT JAS_ATTRIBUTE_PURE int jas_stream_isseekable (jas_stream_t *stream)

  Determine if stream supports seeking.

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

  Set the current position within the stream.
- JAS_EXPORT long jas_stream_tell (jas_stream_t *stream)

  Get the current position within the stream.
- JAS_EXPORT int jas_stream_rewind (jas_stream_t *stream)

  Seek to the beginning of a stream.
- JAS_EXPORT int jas_stream_flush (jas_stream_t *stream)

  Flush any pending output to a stream.
- JAS_EXPORT int jas_stream_copy (jas_stream_t *destination, jas_stream_t *source, ssize_t count)

  Copy data from one stream to another.
- JAS_EXPORT int jas_stream_display (jas_stream_t *stream, FILE *fp, int count)

  Print a hex dump of data read from a stream.
- JAS_EXPORT ssize_t jas_stream_gobble (jas_stream_t *stream, size_t count)

  Consume (i.e., discard) characters from stream.
- JAS_EXPORT ssize_t jas_stream_pad (jas_stream_t *stream, size_t count, int value)

  Write a fill character multiple times to a stream.
- JAS_EXPORT long jas_stream_length (jas_stream_t *stream)

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

14.3.1 Detailed Description

I/O streams.

General information can be found here.

14.3.2 Macro Definition Documentation

14.3.2.1 jas_stream_clearerr

```
#define jas_stream_clearerr(
            stream )  ((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

stream	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

stream	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

stream	A pointer to the stream from which to read a character.

Returns

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

14.3.2.5 jas_stream_getrwcount

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

Get the read/write count for a stream.

Parameters

stream	A pointer to the stream whose read/write count is to be queried.

Returns

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

14.3.2.6 jas_stream_getrwlimit

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

Get the read/write limit for a stream.

Parameters

stream	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

stream	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

stream	A pointer to the stream to which to write the character.
c	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

stream	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

destination	A pointer to the stream that is the destination for the copy.
source	A pointer to the stream that is the source for the copy.
count	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

stream	A pointer to the stream from which to read data.
fp	A pointer to a stdio stream (i.e., FILE) to which to print the hex dump.
count	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

fd	The file descriptor of the file to open as a stream.
mode	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

stream	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

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

Returns

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

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

path	A pointer to a null-terminated string containing the pathname of the file to be reopened.
mode	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 fdopen function in the C standard library.
fp	A pointer to the `FILE` (i.e., stdio stream) to be reopened.

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

Returns

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

14.3.3.8 jas_stream_gets()

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

Read a line of input from a stream.

Parameters

stream	A pointer to the stream from which to read input.
buffer	A pointer to the start of the buffer to hold to input to be read.
buffer_size	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

stream	A pointer to the stream from which to discard data.
count	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

stream	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

stream	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

buffer	A pointer to the buffer to be used to store stream data.
buffer_size	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

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

stream	A pointer to the stream from which to retrieve pending input.
buffer	A pointer to the start of the buffer.
count	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

stream	A pointer to the stream to which to write output.
format	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

stream	A pointer to the stream to which the string should be written.
s	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

stream	A pointer to the stream from which to read data.
buffer	A pointer to the start of the buffer.
count	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_getrwcount() 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

stream	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

stream	A pointer to the stream for which to set the current position.
offset	The new position for the stream.
origin	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_setrwcount()

```
JAS_EXPORT long jas_stream_setrwcount (
            jas_stream_t * stream,
            long rw_count )
```

Set the read/write count for a stream.

Parameters

stream	A pointer to the stream whose read/write count is to be set.
rw_count	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_setrwlimit()

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

Set the read/write limit for a stream.

Parameters

stream	A pointer to the stream whose read/write limit is to be set.
rwlimit	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

stream	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

stream	A pointer to the stream to which the character should be put back.
c	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

stream	A pointer to the stream to which to write data.
buffer	A pointer to the start of the buffer.
count	A count of the number of characters to write.

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

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

Returns

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

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

14.4 Image Representation

Image Representation.

Classes

- struct jas_image_cmpt_t

  Image component class.
- struct jas_image_t

  Image class.
- struct jas_image_cmptparm_t

  Component parameters class.
- struct jas_image_fmtops_t

  Image format-dependent operations.
- struct jas_image_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_cmpthstep(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_cmprof(image) ((image)->cmprof_)

  Get the color management profile of an image.
- #define jas_image_setcmprof(image, cmprof) ((image)->cmprof_ = cmprof)

  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_fmttostr (int fmt)

  Get the name of the image format with the specified ID.
- JAS_ATTRIBUTE_CONST JAS_EXPORT const jas_image_fmtinfo_t * jas_image_lookupfmtbyid (int id)

  Lookup image format information by the format ID.
- JAS_ATTRIBUTE_PURE JAS_EXPORT const jas_image_fmtinfo_t * jas_image_lookupfmtbyname (const char *name)

  Lookup image format information by the format name.
- JAS_ATTRIBUTE_PURE JAS_EXPORT int jas_image_fmtfromname (const char *filename)

  Guess the format of an image file based on its name.
- JAS_ATTRIBUTE_PURE JAS_EXPORT int jas_image_getfmt (jas_stream_t *in)

  Get the format of image data in a stream.
- JAS_ATTRIBUTE_PURE JAS_EXPORT int jas_image_ishomosamp (const jas_image_t *image)

  Test if the sampling of the image is homogeneous.

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

  ???
- JAS_EXPORT int jas_image_writecmpt2 (jas_image_t *image, unsigned cmptno, jas_image_coord_t x, jas_image_coord_t y, jas_image_coord_t width, jas_image_coord_t height, const long *buf)

  Write sample data in a component of an image.
- JAS_EXPORT int jas_image_readcmpt2 (jas_image_t *image, unsigned cmptno, jas_image_coord_t x, jas_image_coord_t y, jas_image_coord_t width, jas_image_coord_t height, long *buf)

  Read sample data in a component of an image.
- JAS_EXPORT jas_image_t * jas_image_chclrspc (jas_image_t *image, const jas_cmprof_t *outprof, jas_cmxform_intent_t intent)

  Change the color space for an image.
- JAS_EXPORT int jas_image_dump (jas_image_t *image, FILE *out)

  Dump the information for an image (for debugging).

14.4.1 Detailed Description

Image Representation.

General information can be found here.

14.4.2 Macro Definition Documentation

14.4.2.1 jas_image_brx

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

```
#define jas_image_cmpthstep(
              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_delcmpt()

```
JAS_EXPORT void jas_image_delcmpt (
            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_fmttostr()

```
JAS_ATTRIBUTE_CONST JAS_EXPORT const char * jas_image_fmttostr (
            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_setfmtenable()

```
JAS_EXPORT int jas_image_setfmtenable (
            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_writecmpt()

```
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.

14.4.4.33 jas_image_writecmpt2()

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

Write sample data in a component of an image.

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_cmxform_t

  Primitive transform class.
- struct jas_cmprof_t

Macros

- #define JAS_CMXFORM_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_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.

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_seqent_t minval,
            jas_seqent_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_seqent_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_seqent_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_seqent_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_seqent_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_ASL JAS_FIX_ASL_FAST
- #define JAS_FIX_ASL_FAST(fix_t, fracbits, x, n) ((x) << (n))
- #define JAS_FIX_ASL_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_ASL

```
#define JAS_FIX_ASL JAS_FIX_ASL_FAST
```

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

14.7.2.7 JAS_FIX_ASL_FAST

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

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

14.7.2.8 JAS_FIX_ASL_OFLOW

```
#define JAS_FIX_ASL_OFLOW(
            fix_t,
            fracbits,
            x,
            n )  ((((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_nonull (const jas_taginfo_t *taginfo)

  Ensure a nonnull taginfo pointer.
- JAS_EXPORT jas_tvparser_t * jas_tvparser_create (const char *s)

  Create a tag-value parser for the specified string.
- JAS_EXPORT void jas_tvparser_destroy (jas_tvparser_t *tvparser)

  Destroy a tag-value parser.
- JAS_EXPORT int jas_tvparser_next (jas_tvparser_t *tvparser)

  Get the next tag-value pair.
- JAS_ATTRIBUTE_PURE JAS_EXPORT const char * jas_tvparser_gettag (const jas_tvparser_t *tvparser)

  Get the tag name for the current tag-value pair.
- JAS_ATTRIBUTE_PURE JAS_EXPORT const char * jas_tvparser_getval (const jas_tvparser_t *tvparser)

  Get the value for the current tag-value pair.

14.11.1 Detailed Description

Tag-Value Pair (TVP) Parsing.

General information can be found here.

14.11.2 Function Documentation

14.11.2.1 jas_taginfo_nonull()

```
JAS_ATTRIBUTE_PURE JAS_EXPORT const jas_taginfo_t * jas_taginfo_nonull (
            const jas_taginfo_t * taginfo )
```

Ensure a nonnull taginfo pointer.

This function returns a pointer to the specified taginfo object if it exists (i.e., the pointer is nonnull); otherwise, a pointer to a dummy object is returned. This is useful in some situations to avoid checking for a null pointer.

14.11.2.2 jas_taginfos_lookup()

```
JAS_ATTRIBUTE_PURE JAS_EXPORT const jas_taginfo_t * jas_taginfos_lookup (
            const jas_taginfo_t * taginfos,
            const char * name )
```

Lookup a tag by name.

14.11.2.3 jas_tvparser_create()

```
JAS_EXPORT jas_tvparser_t * jas_tvparser_create (
            const char * s )
```

Create a tag-value parser for the specified string.

14.11.2.4 jas_tvparser_destroy()

```
JAS_EXPORT void jas_tvparser_destroy (
            jas_tvparser_t * tvparser )
```

Destroy a tag-value parser.

14.11.2.5 jas_tvparser_gettag()

```
JAS_ATTRIBUTE_PURE JAS_EXPORT const char * jas_tvparser_gettag (
            const jas_tvparser_t * tvparser )
```

Get the tag name for the current tag-value pair.

14.11.2.6 jas_tvparser_getval()

```
JAS_ATTRIBUTE_PURE JAS_EXPORT const char * jas_tvparser_getval (
            const jas_tvparser_t * tvparser )
```

Get the value for the current tag-value pair.

14.11.2.7 jas_tvparser_next()

```
JAS_EXPORT int jas_tvparser_next (
            jas_tvparser_t * tvparser )
```

Get the next tag-value pair.

14.12 String Processing

String Processing.

Functions

- JAS_EXPORT char ∗ jas_strdup (const char ∗)

  Create a copy of a null-terminated string.
- JAS_EXPORT char ∗ jas_strtok (char ∗str, const char ∗delim, char ∗∗saveptr)

  Extract tokens from a string.
- JAS_EXPORT int jas_stringtokenize (const char ∗string, const char ∗delim, char ∗∗∗tokens_buf, size_t ∗max_↩ tokens_buf, size_t ∗num_tokens_buf)

  Split a string into tokens based on specified delimiters.

14.12.1 Detailed Description

String Processing.

General information can be found here.

14.12.2 Function Documentation

14.12.2.1 jas_strdup()

```
JAS_EXPORT char * jas_strdup (
            const char * s )
```

Create a copy of a null-terminated string.

This function has a behavior similar to the well-known strdup function.

14.12.2.2 jas_stringtokenize()

```
JAS_EXPORT int jas_stringtokenize (
            const char * string,
            const char * delim,
            char *** tokens_buf,
            size_t * max_tokens_buf,
            size_t * num_tokens_buf )
```

Split a string into tokens based on specified delimiters.

Parameters

string	A pointer to a null-terminated string to be split into tokens.
delim	A pointer to a null-terminated string contained characters used to delimit tokens.
tokens_buf	A pointer to the output token array.
max_tokens_buf	A pointer to the allocated size of the token array.
num_tokens_buf	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

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * Windows Bitmap File Library
 *
 * $Id$
 */

#ifndef BMP_COD_H
#define BMP_COD_H

/*****************************************************************************\
* Includes.
\*****************************************************************************/

#include "jasper/jas_types.h"

/*****************************************************************************\
* Constants and macros.
\*****************************************************************************/

#define BMP_MAGIC       0x4d42
/* The signature for a BMP file. */

#define BMP_HDRLEN      14
/* The nominal header length. */

#define BMP_INFOLEN     40
/* The nominal info length. */

#define BMP_PALLEN(info)        ((info)->numcolors * 4)
/* The length of the palette. */

#define BMP_HASPAL(info)        ((info)->numcolors > 0)
/* Is this a palettized image? */

/* Encoding types. */
#define BMP_ENC_RGB             0 /* No special encoding. */
#define BMP_ENC_RLE8    1 /* Run length encoding. */
#define BMP_ENC_RLE4    2 /* Run length encoding. */

/*****************************************************************************\
* Types.
\*****************************************************************************/

/* BMP header. */
typedef struct {

        int_fast16_t magic;
        /* The signature (a.k.a. the magic number). */

        int_fast32_t siz;
        /* The size of the file in 32-bit words. */

        int_fast16_t reserved1;
        /* Ask Bill Gates what this is all about. */

        int_fast16_t reserved2;
        /* Ditto. */

        int_fast32_t off;
        /* The offset of the bitmap data from the bitmap file header in bytes. */

} bmp_hdr_t;

/* Palette entry. */
typedef struct {

        int_fast16_t red;
        /* The red component. */

```

```
        int_fast16_t grn;
        /* The green component. */

        int_fast16_t blu;
        /* The blue component. */

        int_fast16_t res;
        /* Reserved. */

} bmp_palent_t;

/* BMP info. */
typedef struct {

        int_fast32_t len;
        /* The length of the bitmap information header in bytes. */

        int_fast32_t width;
        /* The width of the bitmap in pixels. */

        int_fast32_t height;
        /* The height of the bitmap in pixels. */

        int_fast8_t topdown;
        /* The bitmap data is specified in top-down order. */

        int_fast16_t numplanes;
        /* The number of planes.  This must be set to a value of one. */

        int_fast16_t depth;
        /* The number of bits per pixel. */

        int_fast32_t enctype;
        /* The type of compression used. */

        int_fast32_t siz;
        /* The size of the image in bytes. */

        int_fast32_t hres;
        /* The horizontal resolution in pixels/metre. */

        int_fast32_t vres;
        /* The vertical resolution in pixels/metre. */

        int_fast32_t numcolors;
        /* The number of color indices used by the bitmap. */

        int_fast32_t mincolors;
        /* The number of color indices important for displaying the bitmap. */

        bmp_palent_t *palents;
        /* The colors should be listed in order of importance. */

} bmp_info_t;

/*****************************************************************************\
* Functions and macros.
\*****************************************************************************/

#define bmp_issupported(hdr, info) \
        ((hdr)->magic == BMP_MAGIC && !(hdr)->reserved1 && \
          !(hdr)->reserved2 && (info)->numplanes == 1 && \
          ((info)->depth == 8 || (info)->depth == 24) && \
          (info)->enctype == BMP_ENC_RGB)
/* Is this type of BMP file supported? */

#define bmp_haspal(info) \
        ((info)->depth == 8)
/* Is there a palette? */

int bmp_numcmpts(bmp_info_t *info);
/* Get the number of components. */

bmp_info_t *bmp_info_create(void);
/* Create BMP information. */

void bmp_info_destroy(bmp_info_t *info);
/* Destroy BMP information. */

int bmp_isgrayscalepal(bmp_palent_t *palents, int numpalents);
/* Does the specified palette correspond to a grayscale image? */
```

```

#endif
```

16.2 bmp_enc.h

```
/*
 * Copyright (c) 2002 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef BMP_ENC_H
#define BMP_ENC_H

typedef struct {

        int numcmpts;
        int cmpts[4];

} bmp_enc_t;

#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_cmxform_t

  Primitive transform class.
- struct jas_cmprof_t

Macros

- #define JAS_CMXFORM_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.

16.3.1 Detailed Description

JasPer Color Management.

16.4 jas_cm.h

Go to the documentation of this file.

```
/*
 * Copyright (c) 2002-2003 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_CM_H
#define JAS_CM_H

/*****************************************************************************\
*
```

```
\*****************************************************************************/

/* The configuration header file should be included first. */
#include <jasper/jas_config.h>

#include <jasper/jas_types.h>
#include <jasper/jas_icc.h>

#ifdef __cplusplus
extern "C" {
#endif

/*****************************************************************************\
* Types and Macros.
\*****************************************************************************/

typedef unsigned jas_clrspc_t;

typedef enum {
        JAS_CMXFORM_OP_FWD = 0,
        JAS_CMXFORM_OP_REV = 1,
        JAS_CMXFORM_OP_PROOF = 2,
        JAS_CMXFORM_OP_GAMUT = 3,
} jas_cmxform_op_t;

typedef enum {
        JAS_CMXFORM_INTENT_PER = 0,
        JAS_CMXFORM_INTENT_RELCLR = 1,
        JAS_CMXFORM_INTENT_ABSCLR = 2,
        JAS_CMXFORM_INTENT_SAT = 3,
} jas_cmxform_intent_t;

#define JAS_CMXFORM_NUMINTENTS          4

typedef enum {
        JAS_CMXFORM_OPTM_SPEED = 0,
        JAS_CMXFORM_OPTM_SIZE = 1,
        JAS_CMXFORM_OPTM_ACC = 2,
} jas_cmxform_optm_t;

#define jas_clrspc_create(fam, mbr)     (((fam) « 8) | (mbr))

#define jas_clrspc_fam(clrspc)  ((clrspc) » 8)

#define jas_clrspc_mbr(clrspc)  ((clrspc) & 0xff)

#define jas_clrspc_isgeneric(clrspc)    (!jas_clrspc_mbr(clrspc))

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

#define JAS_CLRSPC_UNKNOWNMASK  0x4000

#define JAS_CLRSPC_FAM_UNKNOWN  0
#define JAS_CLRSPC_FAM_XYZ      1
#define JAS_CLRSPC_FAM_LAB      2
#define JAS_CLRSPC_FAM_GRAY     3
#define JAS_CLRSPC_FAM_RGB      4
#define JAS_CLRSPC_FAM_YCBCR    5

#define JAS_CLRSPC_UNKNOWN      JAS_CLRSPC_UNKNOWNMASK
#define JAS_CLRSPC_CIEXYZ       jas_clrspc_create(JAS_CLRSPC_FAM_XYZ, 1)
#define JAS_CLRSPC_CIELAB       jas_clrspc_create(JAS_CLRSPC_FAM_LAB, 1)
#define JAS_CLRSPC_SGRAY        jas_clrspc_create(JAS_CLRSPC_FAM_GRAY, 1)
#define JAS_CLRSPC_SRGB         jas_clrspc_create(JAS_CLRSPC_FAM_RGB, 1)
#define JAS_CLRSPC_SYCBCR       jas_clrspc_create(JAS_CLRSPC_FAM_YCBCR, 1)

#define JAS_CLRSPC_GENRGB       jas_clrspc_create(JAS_CLRSPC_FAM_RGB, 0)
#define JAS_CLRSPC_GENGRAY      jas_clrspc_create(JAS_CLRSPC_FAM_GRAY, 0)
#define JAS_CLRSPC_GENYCBCR     jas_clrspc_create(JAS_CLRSPC_FAM_YCBCR, 0)

#define JAS_CLRSPC_CHANIND_YCBCR_Y      0
#define JAS_CLRSPC_CHANIND_YCBCR_CB     1
#define JAS_CLRSPC_CHANIND_YCBCR_CR     2

#define JAS_CLRSPC_CHANIND_RGB_R        0
#define JAS_CLRSPC_CHANIND_RGB_G        1
#define JAS_CLRSPC_CHANIND_RGB_B        2

#define JAS_CLRSPC_CHANIND_GRAY_Y       0

typedef double jas_cmreal_t;
```

```
struct jas_cmpxform_s;

typedef struct {
        long *buf;
        unsigned prec;
        int sgnd;
        unsigned width;
        unsigned height;
} jas_cmcmptfmt_t;

typedef struct {
        unsigned numcmpts;
        jas_cmcmptfmt_t *cmptfmts;
} jas_cmpixmap_t;

typedef struct {
        void (*destroy)(struct jas_cmpxform_s *pxform);
        int (*apply)(const struct jas_cmpxform_s *pxform, const jas_cmreal_t *in, jas_cmreal_t *out,
    unsigned  cnt);
        void (*dump)(struct jas_cmpxform_s *pxform);
} jas_cmpxformops_t;

typedef struct {
        jas_cmreal_t *data;
        unsigned size;
} jas_cmshapmatlut_t;

typedef struct {
        int mono;
        int order;
        int useluts;
        int usemat;
        jas_cmshapmatlut_t luts[3];
        jas_cmreal_t mat[3][4];
} jas_cmshapmat_t;

typedef struct {
        int order;
} jas_cmshaplut_t;

typedef struct {
        unsigned inclrspc;
        unsigned outclrspc;
} jas_cmclrspcconv_t;

typedef struct jas_cmpxform_s {
        unsigned refcnt;
        const jas_cmpxformops_t *ops;
        unsigned numinchans;
        unsigned numoutchans;
        union {
                max_align_t dummy;
                jas_cmshapmat_t shapmat;
                jas_cmshaplut_t shaplut;
                jas_cmclrspcconv_t clrspcconv;
        } data;
} jas_cmpxform_t;

typedef struct {
        unsigned numpxforms;
        unsigned maxpxforms;
        jas_cmpxform_t **pxforms;
} jas_cmpxformseq_t;

typedef struct {
        unsigned numinchans;
        unsigned numoutchans;
        jas_cmpxformseq_t *pxformseq;
} jas_cmxform_t;

#define JAS_CMPROF_TYPE_DEV     1
#define JAS_CMPROF_TYPE_CLRSPC  2

#define JAS_CMPROF_NUMPXFORMSEQS        13

typedef struct {
        jas_clrspc_t clrspc;
        unsigned numchans;
        unsigned refclrspc;
        unsigned numrefchans;
```

```
        jas_iccprof_t *iccprof;
        jas_cmpxformseq_t *pxformseqs[JAS_CMPROF_NUMPXFORMSEQS];
} jas_cmprof_t;

/*****************************************************************************\
*
\*****************************************************************************/

#if 0
typedef int_fast32_t jas_cmattrname_t;
typedef int_fast32_t jas_cmattrval_t;
typedef int_fast32_t jas_cmattrtype_t;
/* Load a profile. */
int jas_cmprof_load(jas_cmprof_t *prof, jas_stream_t *in, unsigned fmt);
/* Save a profile. */
int jas_cmprof_save(jas_cmprof_t *prof, jas_stream_t *out, unsigned fmt);
/* Set an attribute of a profile. */
int jas_cm_prof_setattr(jas_cm_prof_t *prof, jas_cm_attrname_t name, void *val);
/* Get an attribute of a profile. */
void *jas_cm_prof_getattr(jas_cm_prof_t *prof, jas_cm_attrname_t name);
#endif

/*****************************************************************************\
* Color-management (CM) profile class.
\*****************************************************************************/

JAS_EXPORT
jas_cmprof_t *jas_cmprof_createfromiccprof(const jas_iccprof_t *iccprof);

JAS_EXPORT
jas_cmprof_t *jas_cmprof_createfromclrspc(jas_clrspc_t clrspc);

JAS_EXPORT
void jas_cmprof_destroy(jas_cmprof_t *prof);

JAS_EXPORT
jas_cmprof_t *jas_cmprof_copy(const jas_cmprof_t *prof);

JAS_EXPORT
jas_iccprof_t *jas_iccprof_createfromcmprof(const jas_cmprof_t *prof);

/*****************************************************************************\
* Color-Management (CM) Transform.
\*****************************************************************************/

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);

JAS_EXPORT
void jas_cmxform_destroy(jas_cmxform_t *xform);

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

/*****************************************************************************\
* Miscellany.
\*****************************************************************************/

unsigned jas_clrspc_numchans(jas_clrspc_t clrspc);

#define jas_cmprof_clrspc(prof) ((prof)->clrspc)

#ifdef __cplusplus
}
#endif

#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.

```
/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_COMPILER_H
#define JAS_COMPILER_H

/* The configuration header file should be included first. */
#include <jasper/jas_config.h>

#ifdef _MSC_VER
#       ifndef __cplusplus
#               undef inline
#               define inline __inline
#       endif
#endif

#if defined(__GNUC__)
#       define JAS_DEPRECATED __attribute__((deprecated))
#else
#       define JAS_DEPRECATED
#endif

#if defined(__GNUC__)
#       define JAS_ATTRIBUTE_CONST __attribute__((const))
#else
#       define JAS_ATTRIBUTE_CONST
#endif

#if defined(__GNUC__)
#       define JAS_ATTRIBUTE_PURE __attribute__((pure))
#else
#       define JAS_ATTRIBUTE_PURE
#endif

#if defined(__GNUC__)
#       define JAS_FORCE_INLINE inline __attribute__((always_inline))
#else
#       define JAS_FORCE_INLINE inline
#endif

#if defined(__GNUC__)
#       if __GNUC__ > 4 || (__GNUC__ == 4 && __GNUC_MINOR__ >= 5)
#               define JAS_UNREACHABLE() __builtin_unreachable()
#       else
#               define JAS_UNREACHABLE()
#       endif
#elif defined(__clang__)
#               define JAS_UNREACHABLE() __builtin_unreachable()
#elif defined(_MSC_VER)
#       define JAS_UNREACHABLE() __assume(0)
#else
#       define JAS_UNREACHABLE()
#endif

#if defined(__GNUC__)
#       define JAS_LIKELY(x) __builtin_expect (!!(x), 1)
```

```
#else
#       define JAS_LIKELY(x) (x)
#endif

#if defined(__GNUC__)
#       define JAS_UNLIKELY(x) __builtin_expect (!!(x), 0)
#else
#       define JAS_UNLIKELY(x) (x)
#endif

#if defined(__GNUC__) && __GNUC__ >= 6
#       define JAS_ATTRIBUTE_DISABLE_UBSAN \
          __attribute__((no_sanitize_undefined))
#elif defined(__clang__)
#       define JAS_ATTRIBUTE_DISABLE_UBSAN \
          __attribute__((no_sanitize("undefined")))
#else
#       define JAS_ATTRIBUTE_DISABLE_UBSAN
#endif

#ifdef __has_builtin
#define jas_has_builtin(x) __has_builtin(x)
#else
#define jas_has_builtin(x) 0
#endif

#define JAS_UNUSED(x) ((void) x)

#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.

```
/*
 * Copyright (c) 2001-2002 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_DEBUG_H
#define JAS_DEBUG_H

/*****************************************************************************\
* Includes.
\*****************************************************************************/

/* The configuration header file should be included first. */
#include <jasper/jas_config.h>

#include "jasper/jas_init.h"
#include "jasper/jas_debug.h"

#include <stdio.h>
#include <stdarg.h>

#ifdef __cplusplus
extern "C" {
#endif

/*****************************************************************************\
* Macros and functions.
\*****************************************************************************/

/* Output debugging information to standard error provided that the debug
  level is set sufficiently high. */
#if !defined(NDEBUG)
#define JAS_DBGLOG(n, x) \
        ((jas_get_debug_level() >= (n)) ? (jas_eprintf x) : 0)
#else
#define JAS_DBGLOG(n, x)
#endif

#if !defined(NDEBUG)
#define JAS_LOGDEBUGF(n, ...) \
         ((jas_get_debug_level() >= (n)) ? jas_logdebugf((n), __VA_ARGS__) : 0)
#else
#define JAS_LOGDEBUGF(n, ...)
#endif

JAS_EXPORT
void jas_deprecated(const char *fmt, ...);

JAS_DEPRECATED
static inline
int jas_getdbglevel(void)
{
        jas_deprecated("jas_getdbglevel is deprecated\n");
        return jas_get_debug_level();
}

JAS_EXPORT
int jas_setdbglevel(int dbglevel);

JAS_EXPORT
int jas_eprintf(const char *fmt, ...);

JAS_EXPORT
int jas_logprintf(const char *fmt, ...);

JAS_EXPORT
int jas_logerrorf(const char *fmt, ...);

```

```
JAS_EXPORT
int jas_logwarnf(const char *fmt, ...);

JAS_EXPORT
int jas_loginfof(const char *fmt, ...);

JAS_EXPORT
int jas_logdebugf(int priority, const char *fmt, ...);

int jas_logmemdump(const void *data, size_t len);

JAS_EXPORT
int jas_memdump(FILE *out, const void *data, size_t len);

#define JAS_STRINGIFY(x) #x

#define JAS_STRINGIFYX(x) JAS_STRINGIFY(x)

#ifdef __cplusplus
}
#endif

#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.

```
/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_DLL_H
#define JAS_DLL_H

/* The configuration header file should be included first. */
#include <jasper/jas_config.h>

#include <jasper/jas_export_cmake.h>

/* For backward compatibility only. */
#define JAS_DLLEXPORT JAS_EXPORT
/* For backward compatibility only. */
#define JAS_DLLLOCAL JAS_LOCAL

#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_ASL JAS_FIX_ASL_FAST
- #define JAS_FIX_ASL_FAST(fix_t, fracbits, x, n) ((x) << (n))
- #define JAS_FIX_ASL_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)))

16.11.1 Detailed Description

JasPer Fixed-Point Number Class.

16.12 jas_fix.h

Go to the documentation of this file.

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_FIX_H
#define JAS_FIX_H

/*****************************************************************************\
* Includes.
\*****************************************************************************/

/* The configuration header file should be included first. */
#include <jasper/jas_config.h> /* IWYU pragma: keep */

#include <jasper/jas_types.h>

#ifdef __cplusplus
```

```
extern "C" {
#endif

/*****************************************************************************\
* Constants.
\*****************************************************************************/

#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))

/*****************************************************************************\
* Conversion operations.
\*****************************************************************************/

#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))))

/*****************************************************************************\
* Basic arithmetic operations.
* All other arithmetic operations are synthesized from these basic operations.
* There are three macros for each type of arithmetic operation.
* One macro always performs overflow/underflow checking, one never performs
* overflow/underflow checking, and one is generic with its behavior
* depending on compile-time flags.
* Only the generic macros should be invoked directly by application code.
\*****************************************************************************/

#if !defined(DEBUG_OVERFLOW)
#define JAS_FIX_ADD                     JAS_FIX_ADD_FAST
#else
#define JAS_FIX_ADD                     JAS_FIX_ADD_OFLOW
#endif

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

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

#if !defined(DEBUG_OVERFLOW)
#define JAS_FIX_MUL                     JAS_FIX_MUL_FAST
#else
#define JAS_FIX_MUL                     JAS_FIX_MUL_OFLOW
#endif

#define JAS_FIX_MUL_FAST(fix_t, fracbits, bigfix_t, x, y) \
        JAS_CAST(fix_t, (JAS_CAST(bigfix_t, x) * JAS_CAST(bigfix_t, y)) » \
          (fracbits))

#define JAS_FIX_MUL_OFLOW(fix_t, fracbits, bigfix_t, x, y) \
        ((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())

#if !defined(DEBUG_OVERFLOW)
#define JAS_FIX_MULBYINT        JAS_FIX_MULBYINT_FAST
#else
#define JAS_FIX_MULBYINT        JAS_FIX_MULBYINT_OFLOW
#endif

#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)

#if !defined(DEBUG_OVERFLOW)
#define JAS_FIX_DIV                     JAS_FIX_DIV_FAST
#else
#define JAS_FIX_DIV                     JAS_FIX_DIV_UFLOW
#endif

#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)

#if !defined(DEBUG_OVERFLOW)
#define JAS_FIX_NEG                     JAS_FIX_NEG_FAST
#else
#define JAS_FIX_NEG                     JAS_FIX_NEG_OFLOW
#endif

#define JAS_FIX_NEG_FAST(fix_t, fracbits, x) \
        (-(x))

/* Yes, overflow is actually possible for two's complement representations,
  although highly unlikely to occur. */
#define JAS_FIX_NEG_OFLOW(fix_t, fracbits, x) \
        (((x) < 0) ? (-(x) > 0 || JAS_FIX_OFLOW(), -(x)) : (-(x)))

#if !defined(DEBUG_OVERFLOW)
#define JAS_FIX_ASL                     JAS_FIX_ASL_FAST
#else
#define JAS_FIX_ASL                     JAS_FIX_ASL_OFLOW
#endif

#define JAS_FIX_ASL_FAST(fix_t, fracbits, x, n) \
        ((x) « (n))

#define JAS_FIX_ASL_OFLOW(fix_t, fracbits, x, n) \
        ((((x) « (n)) » (n)) == (x) || JAS_FIX_OFLOW(), (x) « (n))

#if !defined(DEBUG_OVERFLOW)
#define JAS_FIX_ASR                     JAS_FIX_ASR_FAST
#else
#define JAS_FIX_ASR                     JAS_FIX_ASR_UFLOW
#endif

#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)

/*****************************************************************************\
* Other basic arithmetic operations.
\*****************************************************************************/

#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))

/*****************************************************************************\
* Miscellaneous operations.
\*****************************************************************************/

#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))

/*****************************************************************************\
* Relational operations.
\*****************************************************************************/

#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))

/*****************************************************************************\
* Rounding functions.
\*****************************************************************************/

#define JAS_FIX_ROUND(fix_t, fracbits, x) \
        (((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)))))

#define JAS_FIX_FLOOR(fix_t, fracbits, x) \
        ((x) & (~(JAS_FIX_ONE(fix_t, fracbits) - 1)))

/*****************************************************************************\
* The below macros are for internal library use only.  Do not invoke them
* directly in application code.
\*****************************************************************************/

/* Handle overflow. */
#define JAS_FIX_OFLOW() \
        jas_logerrorf("overflow error: file %s, line %d\n", __FILE__, __LINE__)

/* Handle underflow. */
#define JAS_FIX_UFLOW() \
        jas_logerrorf("underflow error: file %s, line %d\n", __FILE__, __LINE__)

#ifdef __cplusplus
}
#endif

#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.

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_GETOPT_H
#define JAS_GETOPT_H

#ifdef __cplusplus
extern "C" {
#endif

/* The configuration header file should be included first. */
#include <jasper/jas_config.h>

/******************************************************************************\
* Constants.
\******************************************************************************/

#define JAS_GETOPT_EOF  (-1)
#define JAS_GETOPT_ERR  '?'

#define JAS_OPT_HASARG  0x01    /* option has argument */

/******************************************************************************\
* Types.
\******************************************************************************/

typedef struct {

        /* The unique identifier for this option. */
        int id;

        /* The name of this option. */
        const char *name;

        /* option flags. */
        int flags;

} jas_opt_t;

/******************************************************************************\
* External data.
\******************************************************************************/

JAS_EXPORT
extern int jas_optind;

JAS_EXPORT
extern const char *jas_optarg;

```

```
JAS_EXPORT
extern int jas_opterr;

/*****************************************************************************\
* Prototypes.
\*****************************************************************************/

JAS_EXPORT
int jas_getopt(int argc, char **argv, const jas_opt_t *opts);

#ifdef __cplusplus
}
#endif

#endif
```

16.15 jas_icc.h File Reference

ICC Profile.

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

Functions

- JAS_EXPORT jas_iccprof_t * jas_iccprof_load (jas_stream_t *in)

  Read an ICC profile from a stream.
- JAS_EXPORT int jas_iccprof_save (jas_iccprof_t *prof, jas_stream_t *out)

  Write an ICC profile to a stream.
- JAS_EXPORT void jas_iccprof_destroy (jas_iccprof_t *prof)

  Destroy an ICC profile.
- JAS_ATTRIBUTE_PURE JAS_EXPORT jas_iccattrval_t * jas_iccprof_getattr (const jas_iccprof_t *prof, jas_iccattrname_t name)

  Get an attribute of an ICC profile.
- JAS_EXPORT int jas_iccprof_setattr (jas_iccprof_t *prof, jas_iccattrname_t name, jas_iccattrval_t *val)

  Set an attribute of an ICC profile.
- JAS_EXPORT void jas_iccprof_dump (const jas_iccprof_t *prof, FILE *out)

  Dump an ICC profile to a stream in human-readable format for debugging purposes.
- JAS_EXPORT jas_iccprof_t * jas_iccprof_copy (const jas_iccprof_t *prof)

  Create a copy of an ICC profile.
- JAS_EXPORT int jas_iccprof_gethdr (const jas_iccprof_t *prof, jas_icchdr_t *hdr)

  Get the header for an ICC profile.
- JAS_EXPORT int jas_iccprof_sethdr (jas_iccprof_t *prof, const jas_icchdr_t *hdr)

  Set the header for an ICC profile.
- JAS_EXPORT void jas_iccattrval_destroy (jas_iccattrval_t *attrval)

  Destroy an ICC profile attribute.
- JAS_EXPORT int jas_iccattrval_allowmodify (jas_iccattrval_t **attrval)

  TODO/FIXME.

- JAS_EXPORT jas_iccattrval_t * jas_iccattrval_clone (jas_iccattrval_t *attrval)

  Create a copy of an ICC profile attribute.
- JAS_EXPORT jas_iccattrval_t * jas_iccattrval_create (jas_iccuint32_t type)

  Create an ICC profile attribute.
- JAS_EXPORT void jas_iccattrtab_dump (const jas_iccattrtab_t *attrtab, FILE *out)

  Dump an ICC profile attribute to a stream in human-readable format for debugging purposes.
- JAS_EXPORT jas_iccprof_t * jas_iccprof_createfrombuf (const jas_uchar *buf, unsigned len)

  Create an ICC profile from a buffer in memory.
- JAS_EXPORT jas_iccprof_t * jas_iccprof_createfromclrspc (unsigned clrspc)

  Create an ICC profile from a color space.

16.15.1 Detailed Description

ICC Profile.

16.16 jas_icc.h

Go to the documentation of this file.

```
/*
 * Copyright (c) 2002-2003 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_ICC_H
#define JAS_ICC_H

/* The configuration header file should be included first. */
#include <jasper/jas_config.h>

#include <jasper/jas_types.h>
#include <jasper/jas_stream.h>

#include <stdio.h>

#ifdef __cplusplus
extern "C" {
#endif

/* Profile file signature. */
#define JAS_ICC_MAGIC           0x61637370

#define JAS_ICC_HDRLEN  128

/* Profile/device class signatures. */
#define JAS_ICC_CLAS_IN 0x73636e72 /* input device */
#define JAS_ICC_CLAS_DPY        0x6d6e7472 /* display device */
#define JAS_ICC_CLAS_OUT        0x70727472 /* output device */
#define JAS_ICC_CLAS_LNK        0x6c696e6b /* device link */
#define JAS_ICC_CLAS_CNV        0x73706163 /* color space conversion */
#define JAS_ICC_CLAS_ABS        0x61627374 /* abstract */
#define JAS_ICC_CLAS_NAM        0x6e6d636c /* named color */

/* Color space signatures. */
#define JAS_ICC_COLORSPC_XYZ    0x58595a20 /* XYZ */
#define JAS_ICC_COLORSPC_LAB    0x4c616220 /* LAB */
#define JAS_ICC_COLORSPC_LUV    0x4c757620 /* LUV */
#define JAS_ICC_COLORSPC_YCBCR  0x59436272 /* YCbCr */
#define JAS_ICC_COLORSPC_YXY    0x59787920 /* Yxy */
#define JAS_ICC_COLORSPC_RGB    0x52474220 /* RGB */
#define JAS_ICC_COLORSPC_GRAY   0x47524159 /* Gray */
#define JAS_ICC_COLORSPC_HSV    0x48535620 /* HSV */
#define JAS_ICC_COLORSPC_HLS    0x484c5320 /* HLS */
#define JAS_ICC_COLORSPC_CMYK   0x434d594b /* CMYK */
#define JAS_ICC_COLORSPC_CMY    0x434d5920 /* CMY */
#define JAS_ICC_COLORSPC_2      0x32434c52 /* 2 channel color */
#define JAS_ICC_COLORSPC_3      0x33434c52 /* 3 channel color */
#define JAS_ICC_COLORSPC_4      0x34434c52 /* 4 channel color */
#define JAS_ICC_COLORSPC_5      0x35434c52 /* 5 channel color */
#define JAS_ICC_COLORSPC_6      0x36434c52 /* 6 channel color */
#define JAS_ICC_COLORSPC_7      0x37434c52 /* 7 channel color */
#define JAS_ICC_COLORSPC_8      0x38434c52 /* 8 channel color */
#define JAS_ICC_COLORSPC_9      0x39434c52 /* 9 channel color */
#define JAS_ICC_COLORSPC_10     0x41434c52 /* 10 channel color */
#define JAS_ICC_COLORSPC_11     0x42434c52 /* 11 channel color */
#define JAS_ICC_COLORSPC_12     0x43434c52 /* 12 channel color */
#define JAS_ICC_COLORSPC_13     0x44434c52 /* 13 channel color */
#define JAS_ICC_COLORSPC_14     0x45434c52 /* 14 channel color */
#define JAS_ICC_COLORSPC_15     0x46434c52 /* 15 channel color */

/* Profile connection color space (PCS) signatures. */
#define JAS_ICC_REFCOLORSPC_XYZ         0x58595a20 /* CIE XYZ */
#define JAS_ICC_REFCOLORSPC_LAB         0x4c616220 /* CIE Lab */

/* Primary platform signatures. */
#define JAS_ICC_PLATFORM_APPL   0x4150504c /* Apple Computer */
#define JAS_ICC_PLATFORM_MSFT   0x4d534654 /* Microsoft */
#define JAS_ICC_PLATFORM_SGI    0x53474920 /* Silicon Graphics */
#define JAS_ICC_PLATFORM_SUNW   0x53554e57 /* Sun Microsystems */
```

```
#define JAS_ICC_PLATFORM_TGNT   0x54474e54 /* Taligent */

/* Profile flags. */
#define JAS_ICC_FLAGS_EMBED     0x01 /* embedded */
#define JAS_ICC_FLAGS_NOSEP     0x02 /* no separate use */

/* Attributes. */
#define JAS_ICC_ATTR_TRANS      0x01 /* transparent */
#define JAS_ICC_ATTR_MATTE      0x02 /* matte */

/* Rendering intents. */
#define JAS_ICC_INTENT_PER      0 /* perceptual */
#define JAS_ICC_INTENT_REL      1 /* relative colorimetric */
#define JAS_ICC_INTENT_SAT      2 /* saturation */
#define JAS_ICC_INTENT_ABS      3 /* absolute colorimetric */

/* Tag signatures. */
#define JAS_ICC_TAG_ATOB0               0x41324230 /* */
#define JAS_ICC_TAG_ATOB1               0x41324231 /* */
#define JAS_ICC_TAG_ATOB2               0x41324232 /* */
#define JAS_ICC_TAG_BLUMATCOL           0x6258595a /* */
#define JAS_ICC_TAG_BLUTRC              0x62545243 /* */
#define JAS_ICC_TAG_BTOA0               0x42324130 /* */
#define JAS_ICC_TAG_BTOA1               0x42324131 /* */
#define JAS_ICC_TAG_BTOA2               0x42324132 /* */
#define JAS_ICC_TAG_CALTIME             0x63616c74 /* */
#define JAS_ICC_TAG_CHARTARGET          0x74617267 /* */
#define JAS_ICC_TAG_CPYRT               0x63707274 /* */
#define JAS_ICC_TAG_CRDINFO             0x63726469 /* */
#define JAS_ICC_TAG_DEVMAKERDESC        0x646d6e64 /* */
#define JAS_ICC_TAG_DEVMODELDESC        0x646d6464 /* */
#define JAS_ICC_TAG_DEVSET              0x64657673 /* */
#define JAS_ICC_TAG_GAMUT               0x67616d74 /* */
#define JAS_ICC_TAG_GRYTRC              0x6b545243 /* */
#define JAS_ICC_TAG_GRNMATCOL           0x6758595a /* */
#define JAS_ICC_TAG_GRNTRC              0x67545243 /* */
#define JAS_ICC_TAG_LUM                 0x6c756d69 /* */
#define JAS_ICC_TAG_MEASURE             0x6d656173 /* */
#define JAS_ICC_TAG_MEDIABLKPT          0x626b7074 /* */
#define JAS_ICC_TAG_MEDIAWHIPT          0x77747074 /* */
#define JAS_ICC_TAG_NAMCOLR             0x6e636f6c /* */
#define JAS_ICC_TAG_NAMCOLR2            0x6e636c32 /* */
#define JAS_ICC_TAG_OUTRESP             0x72657370 /* */
#define JAS_ICC_TAG_PREVIEW0            0x70726530 /* */
#define JAS_ICC_TAG_PREVIEW1            0x70726531 /* */
#define JAS_ICC_TAG_PREVIEW2            0x70726532 /* */
#define JAS_ICC_TAG_PROFDESC            0x64657363 /* */
#define JAS_ICC_TAG_PROFSEQDESC         0x70736571 /* */
#define JAS_ICC_TAG_PSDCRD0             0x70736430 /* */
#define JAS_ICC_TAG_PSCRDD1             0x70736431 /* */
#define JAS_ICC_TAG_PSCRDD2             0x70736432 /* */
#define JAS_ICC_TAG_PSCRDD3             0x70736433 /* */
#define JAS_ICC_TAG_PS2CSA              0x70733273 /* */
#define JAS_ICC_TAG_PS2RENINTENT        0x70733269 /* */
#define JAS_ICC_TAG_REDMATCOL           0x7258595a /* */
#define JAS_ICC_TAG_REDTRC              0x72545243 /* */
#define JAS_ICC_TAG_SCRNGDES            0x73637264 /* */
#define JAS_ICC_TAG_SCRNG               0x7363726e /* */
#define JAS_ICC_TAG_TECH                0x74656368 /* */
#define JAS_ICC_TAG_UCRBG               0x62666420 /* */
#define JAS_ICC_TAG_VIEWCONDDESC        0x76756564 /* */
#define JAS_ICC_TAG_VIEWCOND            0x76696577 /* */

/* Type signatures. */
#define JAS_ICC_TYPE_CRDINFO            0x63726469 /* CRD information */
#define JAS_ICC_TYPE_CURV               0x63757276 /* curve */
#define JAS_ICC_TYPE_DATA               0x64617461 /* data */
#define JAS_ICC_TYPE_TIME               0x6474696d /* date/time */
#define JAS_ICC_TYPE_DEVSET             0x64657673 /* device settings */
#define JAS_ICC_TYPE_LUT16              0x6d667432 /* */
#define JAS_ICC_TYPE_LUT8               0x6d667431 /* */
#define JAS_ICC_TYPE_MEASURE            0x6d656173 /* */
#define JAS_ICC_TYPE_NAMCOLR            0x6e636f6c /* */
#define JAS_ICC_TYPE_NAMCOLR2           0x6e636c32 /* */
#define JAS_ICC_TYPE_PROFSEQDESC        0x70736571 /* profile sequence description */
#define JAS_ICC_TYPE_RESPCURVSET16      0x72637332 /* response curve set 16 */
#define JAS_ICC_TYPE_SF32               0x73663332 /* signed 32-bit fixed-point */
#define JAS_ICC_TYPE_SCRNG              0x7363726e /* screening */
#define JAS_ICC_TYPE_SIG                0x73696720 /* signature */
#define JAS_ICC_TYPE_TXTDESC            0x64657363 /* text description */
#define JAS_ICC_TYPE_TXT                0x74657874 /* text */
```

```
#define JAS_ICC_TYPE_UF32               0x75663332 /* unsigned 32-bit fixed-point */
#define JAS_ICC_TYPE_UCRBG              0x62666420 /* */
#define JAS_ICC_TYPE_UI16               0x75693136 /* */
#define JAS_ICC_TYPE_UI32               0x75693332 /* */
#define JAS_ICC_TYPE_UI8                0x75693038 /* */
#define JAS_ICC_TYPE_UI64               0x75693634 /* */
#define JAS_ICC_TYPE_VIEWCOND           0x76696577 /* */
#define JAS_ICC_TYPE_XYZ                0x58595a20 /* XYZ */

typedef uint_fast8_t jas_iccuint8_t;
typedef uint_fast16_t jas_iccuint16_t;
typedef uint_fast32_t jas_iccuint32_t;
typedef int_fast32_t jas_iccsint32_t;
typedef int_fast32_t jas_iccs15fixed16_t;
typedef uint_fast32_t jas_iccu16fixed16_t;
typedef uint_fast64_t jas_iccuint64_t;
typedef uint_fast32_t jas_iccsig_t;

typedef jas_iccsig_t jas_icctagsig_t;
typedef jas_iccsig_t jas_icctagtype_t;
typedef jas_iccsig_t jas_iccattrname_t;

/* Date/time type. */
typedef struct {
        jas_iccuint16_t year;
        jas_iccuint16_t month;
        jas_iccuint16_t day;
        jas_iccuint16_t hour;
        jas_iccuint16_t min;
        jas_iccuint16_t sec;
} jas_icctime_t;

/* XYZ type. */
typedef struct {
        jas_iccs15fixed16_t x;
        jas_iccs15fixed16_t y;
        jas_iccs15fixed16_t z;
} jas_iccxyz_t;

/* Curve type. */
typedef struct {
        jas_iccuint32_t numents;
        jas_iccuint16_t *ents;
} jas_icccurv_t;

/* Text description type. */
typedef struct {
        jas_iccuint32_t asclen;
        char *ascdata; /* ASCII invariant description */
        jas_iccuint32_t uclangcode; /* Unicode language code */
        jas_iccuint32_t uclen; /* Unicode localizable description count */
        jas_uchar *ucdata; /* Unicode localizable description */
        jas_iccuint16_t sccode; /* ScriptCode code */
        jas_iccuint8_t maclen; /* Localizable Macintosh description count */
        jas_uchar macdata[69]; /* Localizable Macintosh description */
} jas_icctxtdesc_t;

/* Text type. */
typedef struct {
        char *string;   /* ASCII character string */
} jas_icctxt_t;

typedef struct {
        jas_iccuint8_t numinchans;
        jas_iccuint8_t numoutchans;
        jas_iccsint32_t e[3][3];
        jas_iccuint8_t clutlen;
        jas_iccuint8_t *clut;
        jas_iccuint16_t numintabents;
        jas_iccuint8_t **intabs;
        jas_iccuint8_t *intabsbuf;
        jas_iccuint16_t numouttabents;
        jas_iccuint8_t **outtabs;
        jas_iccuint8_t *outtabsbuf;
} jas_icclut8_t;

typedef struct {
        jas_iccuint8_t numinchans;
        jas_iccuint8_t numoutchans;
        jas_iccsint32_t e[3][3];
        jas_iccuint8_t clutlen;
```

```
        jas_iccuint16_t *clut;
        jas_iccuint16_t numintabents;
        jas_iccuint16_t **intabs;
        jas_iccuint16_t *intabsbuf;
        jas_iccuint16_t numouttabents;
        jas_iccuint16_t **outtabs;
        jas_iccuint16_t *outtabsbuf;
} jas_icclut16_t;

struct jas_iccattrval_s;

typedef struct {
        void (*destroy)(struct jas_iccattrval_s *);
        int (*copy)(struct jas_iccattrval_s *, const struct jas_iccattrval_s *);
        int (*input)(struct jas_iccattrval_s *, jas_stream_t *, unsigned);
//#ifdef JAS_ENABLE_ENCODER
        int (*output)(struct jas_iccattrval_s *, jas_stream_t *);
//#endif
        unsigned (*getsize)(const struct jas_iccattrval_s *);
        void (*dump)(const struct jas_iccattrval_s *, FILE *);
} jas_iccattrvalops_t;

/* Attribute value type (type and value information). */
typedef struct jas_iccattrval_s {
        unsigned refcnt; /* reference count */
        jas_iccsig_t type; /* type */
        const jas_iccattrvalops_t *ops; /* type-dependent operations */
        union {
                jas_iccxyz_t xyz;
                jas_icccurv_t curv;
                jas_icctxtdesc_t txtdesc;
                jas_icctxt_t txt;
                jas_icclut8_t lut8;
                jas_icclut16_t lut16;
        } data; /* value */
} jas_iccattrval_t;

/* Header type. */
typedef struct {
        jas_iccuint32_t size; /* profile size */
        jas_iccsig_t cmmtype; /* CMM type signature */
        jas_iccuint32_t version; /* profile version */
        jas_iccsig_t clas; /* profile/device class signature */
        jas_iccsig_t colorspc; /* color space of data */
        jas_iccsig_t refcolorspc; /* profile connection space */
        jas_icctime_t ctime; /* creation time */
        jas_iccsig_t magic; /* profile file signature */
        jas_iccsig_t platform; /* primary platform */
        jas_iccuint32_t flags; /* profile flags */
        jas_iccsig_t maker; /* device manufacturer signature */
        jas_iccsig_t model; /* device model signature */
        jas_iccuint64_t attr; /* device setup attributes */
        jas_iccsig_t intent; /* rendering intent */
        jas_iccxyz_t illum; /* illuminant */
        jas_iccsig_t creator; /* profile creator signature */
} jas_icchdr_t;

typedef struct {
        jas_iccsig_t name;
        jas_iccattrval_t *val;
} jas_iccattr_t;

typedef struct {
        unsigned numattrs;
        unsigned maxattrs;
        jas_iccattr_t *attrs;
} jas_iccattrtab_t;

typedef struct jas_icctagtabent_s {
        jas_iccuint32_t tag;
        jas_iccuint32_t off;
        jas_iccuint32_t len;
        void *data;
        struct jas_icctagtabent_s *first;
} jas_icctagtabent_t;

typedef struct {
        jas_iccuint32_t numents;
        jas_icctagtabent_t *ents;
} jas_icctagtab_t;

```

```
/* ICC profile type. */
typedef struct {
        jas_icchdr_t hdr;
        jas_icctagtab_t tagtab;
        jas_iccattrtab_t *attrtab;
} jas_iccprof_t;

typedef struct {
        jas_iccuint32_t type;
        jas_iccattrvalops_t ops;
} jas_iccattrvalinfo_t;

JAS_EXPORT
jas_iccprof_t *jas_iccprof_load(jas_stream_t *in);

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

JAS_EXPORT
void jas_iccprof_destroy(jas_iccprof_t *prof);

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

JAS_EXPORT
int jas_iccprof_setattr(jas_iccprof_t *prof, jas_iccattrname_t name,
  jas_iccattrval_t *val);

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

JAS_EXPORT
jas_iccprof_t *jas_iccprof_copy(const jas_iccprof_t *prof);

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

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

JAS_EXPORT
void jas_iccattrval_destroy(jas_iccattrval_t *attrval);

JAS_EXPORT
void jas_iccattrval_dump(const jas_iccattrval_t *attrval, FILE *out);

JAS_EXPORT
int jas_iccattrval_allowmodify(jas_iccattrval_t **attrval);

JAS_EXPORT
jas_iccattrval_t *jas_iccattrval_clone(jas_iccattrval_t *attrval);

JAS_EXPORT
jas_iccattrval_t *jas_iccattrval_create(jas_iccuint32_t type);

JAS_EXPORT
void jas_iccattrtab_dump(const jas_iccattrtab_t *attrtab, FILE *out);

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

JAS_EXPORT
jas_iccprof_t *jas_iccprof_createfromclrspc(unsigned clrspc);

JAS_EXPORT
extern const jas_uchar jas_iccprofdata_srgb[];

JAS_EXPORT
extern const unsigned jas_iccprofdata_srgblen;

JAS_EXPORT
extern const jas_uchar jas_iccprofdata_sgray[];

JAS_EXPORT
extern const unsigned jas_iccprofdata_sgraylen;

#ifdef __cplusplus
}
#endif
```

```
#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_cmpthstep(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_cmprof(image) ((image)->cmprof_)

  Get the color management profile of an image.
- #define jas_image_setcmprof(image, cmprof) ((image)->cmprof_ = cmprof)

  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_fmttostr (int fmt)

  Get the name of the image format with the specified ID.
- JAS_ATTRIBUTE_CONST JAS_EXPORT const jas_image_fmtinfo_t * jas_image_lookupfmtbyid (int id)

  Lookup image format information by the format ID.
- JAS_ATTRIBUTE_PURE JAS_EXPORT const jas_image_fmtinfo_t * jas_image_lookupfmtbyname (const char *name)

  Lookup image format information by the format name.
- JAS_ATTRIBUTE_PURE JAS_EXPORT int jas_image_fmtfromname (const char *filename)

  Guess the format of an image file based on its name.
- JAS_ATTRIBUTE_PURE JAS_EXPORT int jas_image_getfmt (jas_stream_t *in)

  Get the format of image data in a stream.
- JAS_ATTRIBUTE_PURE JAS_EXPORT int jas_image_ishomosamp (const jas_image_t *image)

  Test if the sampling of the image is homogeneous.
- JAS_EXPORT int jas_image_sampcmpt (jas_image_t *image, unsigned cmptno, unsigned newcmptno, jas_image_coord_t ho, jas_image_coord_t vo, jas_image_coord_t hs, jas_image_coord_t vs, int sgnd, unsigned prec)

  ???
- JAS_EXPORT int jas_image_writecmpt2 (jas_image_t *image, unsigned cmptno, jas_image_coord_t x, jas_image_coord_t y, jas_image_coord_t width, jas_image_coord_t height, const long *buf)

  Write sample data in a component of an image.
- JAS_EXPORT int jas_image_readcmpt2 (jas_image_t *image, unsigned cmptno, jas_image_coord_t x, jas_image_coord_t y, jas_image_coord_t width, jas_image_coord_t height, long *buf)

  Read sample data in a component of an image.
- JAS_EXPORT jas_image_t * jas_image_chclrspc (jas_image_t *image, const jas_cmprof_t *outprof, jas_cmxform_intent_t intent)

  Change the color space for an image.
- JAS_EXPORT int jas_image_dump (jas_image_t *image, FILE *out)

  Dump the information for an image (for debugging).

16.17.1 Detailed Description

JasPer Image Class.

16.18 jas_image.h

Go to the documentation of this file.

```
/*
 * Copyright (c) 1999-2000 Image Power, Inc. and the University of
 *   British Columbia.
 * Copyright (c) 2001-2003 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_IMAGE_H
#define JAS_IMAGE_H

/*****************************************************************************\
* Includes.
\*****************************************************************************/

/* The configuration header file should be included first. */
#include <jasper/jas_config.h>

#include <jasper/jas_stream.h>
#include <jasper/jas_types.h>
#include <jasper/jas_seq.h> /* IWYU pragma: export */
```

```
#include <jasper/jas_cm.h> /* IWYU pragma: export */
#include <stdio.h>

#ifdef __cplusplus
extern "C" {
#endif

/******************************************************************************\
* Constants.
\******************************************************************************/

/*
 * Miscellaneous constants.
 */

/* Basic units */
#define JAS_IMAGE_KIBI                  (JAS_CAST(size_t, 1024))
#define JAS_IMAGE_MEBI                  (JAS_IMAGE_KIBI * JAS_IMAGE_KIBI)

/* The threshold at which image data is no longer stored in memory. */
#define JAS_IMAGE_INMEMTHRESH   (256 * JAS_IMAGE_MEBI)

/*
 * Component types
 */

#define JAS_IMAGE_CT_UNKNOWN    0x10000
#define JAS_IMAGE_CT_COLOR(n)   ((n) & 0x7fff)
#define JAS_IMAGE_CT_OPACITY    0x08000

#define JAS_IMAGE_CT_RGB_R      0
#define JAS_IMAGE_CT_RGB_G      1
#define JAS_IMAGE_CT_RGB_B      2

#define JAS_IMAGE_CT_YCBCR_Y    0
#define JAS_IMAGE_CT_YCBCR_CB   1
#define JAS_IMAGE_CT_YCBCR_CR   2

#define JAS_IMAGE_CT_GRAY_Y     0

/******************************************************************************\
* Simple types.
\******************************************************************************/

typedef int_fast32_t jas_image_coord_t;
#define JAS_IMAGE_COORD_MAX INT_FAST32_MAX
#define JAS_IMAGE_COORD_MIN INT_FAST32_MIN

typedef int_fast16_t jas_image_colorspc_t;

typedef int_fast32_t jas_image_cmpttype_t;

typedef int_fast16_t jas_image_smpltype_t;

/******************************************************************************\
* Image class and supporting classes.
\******************************************************************************/

typedef struct {

        /* The x-coordinate of the top-left corner of the component. */
        jas_image_coord_t tlx_;

        /* The y-coordinate of the top-left corner of the component. */
        jas_image_coord_t tly_;

        /* The horizontal sampling period in units of the reference grid. */
        jas_image_coord_t hstep_;

        /* The vertical sampling period in units of the reference grid. */
        jas_image_coord_t vstep_;

        /* The component width in samples. */
        jas_image_coord_t width_;

        /* The component height in samples. */
        jas_image_coord_t height_;

        /* The precision of the sample data (i.e., the number of bits per sample).
        If the samples are signed values, this quantity includes the sign bit. */
        unsigned prec_;
```

```

        /* The signedness of the sample data. */
        int sgnd_;

        /* The stream containing the component data. */
        jas_stream_t *stream_;

        /* The number of characters per sample in the stream. */
        unsigned cps_;

        /* The type of component (e.g., opacity, red, green, blue, luma). */
        jas_image_cmpttype_t type_;

} jas_image_cmpt_t;

typedef struct {

        /* The x-coordinate of the top-left corner of the image bounding box. */
        jas_image_coord_t tlx_;

        /* The y-coordinate of the top-left corner of the image bounding box. */
        jas_image_coord_t tly_;

        /* The x-coordinate of the bottom-right corner of the image bounding
          box (plus one). */
        jas_image_coord_t brx_;

        /* The y-coordinate of the bottom-right corner of the image bounding
          box (plus one). */
        jas_image_coord_t bry_;

        /* The number of components. */
        unsigned numcmpts_;

        /* The maximum number of components that this image can have (i.e., the
          allocated size of the components array). */
        unsigned maxcmpts_;

        /* Per-component information. */
        jas_image_cmpt_t **cmpts_;

        /* The color space. */
        jas_clrspc_t clrspc_;

        /* The CM profile. */
        jas_cmprof_t *cmprof_;

        //bool inmem_;

} jas_image_t;

typedef struct {

        /* The x-coordinate of the top-left corner of the component. */
        jas_image_coord_t tlx;

        /* The y-coordinate of the top-left corner of the component. */
        jas_image_coord_t tly;

        /* The horizontal sampling period in units of the reference grid. */
        jas_image_coord_t hstep;

        /* The vertical sampling period in units of the reference grid. */
        jas_image_coord_t vstep;

        /* The width of the component in samples. */
        jas_image_coord_t width;

        /* The height of the component in samples. */
        jas_image_coord_t height;

        /* The precision of the component sample data. */
        unsigned prec;

        /* The signedness of the component sample data. */
        int sgnd;

} jas_image_cmptparm_t;

/*****************************************************************************\
* File format related classes.
```

```
\*****************************************************************************/

#define JAS_IMAGE_MAXFMTS       32

typedef struct {

        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);

} jas_image_fmtops_t;

typedef struct {

        int id;

        char *name;

        /* The primary file name extension associated with this format. */
        /* This member only exists for backward compatibility. */
        char *ext;

        char **exts;
        size_t max_exts;
        size_t num_exts;

        int enabled;

        char *desc;

        jas_image_fmtops_t ops;

} jas_image_fmtinfo_t;

/*****************************************************************************\
* Image operations.
\*****************************************************************************/

JAS_EXPORT
jas_image_t *jas_image_create(unsigned numcmpts,
  const jas_image_cmptparm_t *cmptparms, jas_clrspc_t clrspc);

JAS_EXPORT
jas_image_t *jas_image_create0(void);

JAS_EXPORT
jas_image_t *jas_image_copy(jas_image_t *image);

JAS_EXPORT
void jas_image_destroy(jas_image_t *image);

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

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

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

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

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

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

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

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

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

#define jas_image_cmpttype(image, cmptno) \
```

```
        ((image)->cmpts_[(cmptno)]->type_)
#define jas_image_setcmpttype(image, cmptno, type) \
        ((image)->cmpts_[(cmptno)]->type_ = (type))

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

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

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

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

#define jas_image_cmpthstep(image, cmptno) \
        ((image)->cmpts_[cmptno]->hstep_)

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

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

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

#define jas_image_cmptbrx(image, cmptno) \
        ((image)->cmpts_[cmptno]->tlx_ + (image)->cmpts_[cmptno]->width_ * \
          (image)->cmpts_[cmptno]->hstep_)

#define jas_image_cmptbry(image, cmptno) \
        ((image)->cmpts_[cmptno]->tly_ + (image)->cmpts_[cmptno]->height_ * \
          (image)->cmpts_[cmptno]->vstep_)

JAS_ATTRIBUTE_PURE
JAS_EXPORT
bool jas_image_cmpt_domains_same(const jas_image_t *image);

JAS_ATTRIBUTE_PURE
JAS_EXPORT
uint_fast32_t jas_image_rawsize(const jas_image_t *image);

JAS_EXPORT
jas_image_t *jas_image_decode(jas_stream_t *in, int fmt, const char *optstr);

JAS_EXPORT
int jas_image_encode(jas_image_t *image, jas_stream_t *out, int fmt,
  const char *optstr);

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);

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);

JAS_EXPORT
void jas_image_delcmpt(jas_image_t *image, unsigned cmptno);

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

JAS_EXPORT
int jas_image_copycmpt(jas_image_t *dstimage, unsigned dstcmptno,
  jas_image_t *srcimage, unsigned srccmptno);

JAS_ATTRIBUTE_CONST
static inline bool JAS_IMAGE_CDT_GETSGND(uint_least8_t dtype)
{
        return (dtype » 7) & 1;
}

JAS_ATTRIBUTE_CONST
static inline uint_least8_t JAS_IMAGE_CDT_SETSGND(bool sgnd)
{
```

```
        return (uint_least8_t)sgnd « 7;
}

JAS_ATTRIBUTE_CONST
static inline uint_least8_t JAS_IMAGE_CDT_GETPREC(uint_least8_t dtype)
{
        return dtype & 0x7f;
}

JAS_ATTRIBUTE_CONST
static inline uint_least8_t JAS_IMAGE_CDT_SETPREC(uint_least8_t dtype)
{
        return dtype & 0x7f;
}

JAS_ATTRIBUTE_PURE
static inline uint_least8_t jas_image_cmptdtype(const jas_image_t *image,
  unsigned cmptno)
{
        return JAS_IMAGE_CDT_SETSGND(image->cmpts_[cmptno]->sgnd_) |
                JAS_IMAGE_CDT_SETPREC(image->cmpts_[cmptno]->prec_);
}

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

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

JAS_EXPORT
void jas_image_writecmptsample(jas_image_t *image, unsigned cmptno,
  unsigned x, unsigned y, int_fast32_t v);

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

/*****************************************************************************\
* Image format-related operations.
\*****************************************************************************/

JAS_EXPORT
void jas_image_clearfmts(void);

#if defined(JAS_FOR_INTERNAL_USE_ONLY)
void jas_image_clearfmts_internal(jas_image_fmtinfo_t *image_fmtinfos,
  size_t *image_numfmts);
#endif

JAS_EXPORT
const jas_image_fmtinfo_t *jas_image_getfmtbyind(int index);

JAS_EXPORT
int jas_image_getnumfmts(void);

#if 0
JAS_EXPORT
int jas_image_delfmtbyid(int id);
#endif

JAS_EXPORT
int jas_image_setfmtenable(int index, int enabled);

#if 0
// TODO: should this be added?
JAS_EXPORT
int jas_image_getfmtindbyname(const char* name);
#endif

JAS_EXPORT
int jas_image_addfmt(int id, const char *name, const char *ext,
  const char *desc, const jas_image_fmtops_t *ops);

#if defined(JAS_FOR_INTERNAL_USE_ONLY)
int jas_image_addfmt_internal(jas_image_fmtinfo_t *image_fmtinfos,
  size_t *image_numfmts, int id, const char *name, const char *ext,
  const char *desc, const jas_image_fmtops_t *ops);
#endif
```

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

JAS_ATTRIBUTE_CONST
JAS_EXPORT
const char *jas_image_fmttostr(int fmt);

JAS_ATTRIBUTE_CONST
JAS_EXPORT
const jas_image_fmtinfo_t *jas_image_lookupfmtbyid(int id);

JAS_ATTRIBUTE_PURE
JAS_EXPORT
const jas_image_fmtinfo_t *jas_image_lookupfmtbyname(const char *name);

JAS_ATTRIBUTE_PURE
JAS_EXPORT
int jas_image_fmtfromname(const char *filename);

JAS_ATTRIBUTE_PURE
JAS_EXPORT
int jas_image_getfmt(jas_stream_t *in);

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

JAS_ATTRIBUTE_PURE
JAS_EXPORT
int jas_image_ishomosamp(const jas_image_t *image);

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

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

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);

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

JAS_EXPORT
jas_image_t *jas_image_chclrspc(jas_image_t *image,
  const jas_cmprof_t *outprof, jas_cmxform_intent_t intent);

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

/*****************************************************************************\
* Image format-dependent operations.
\*****************************************************************************/

#if defined(JAS_INCLUDE_JPG_CODEC)
/* Format-dependent operations for JPG support. */
//JAS_EXPORT
jas_image_t *jpg_decode(jas_stream_t *in, const char *optstr);
//JAS_EXPORT
int jpg_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
//JAS_EXPORT
int jpg_validate(jas_stream_t *in);
#endif

#if defined(JAS_INCLUDE_HEIC_CODEC)
/* Format-dependent operations for HEIC support. */
//JAS_EXPORT
jas_image_t *jas_heic_decode(jas_stream_t *in, const char *optstr);
//JAS_EXPORT
int jas_heic_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
//JAS_EXPORT
int jas_heic_validate(jas_stream_t *in);
#endif

#if defined(JAS_INCLUDE_MIF_CODEC)
/* Format-dependent operations for MIF support. */
```

```
//JAS_EXPORT
jas_image_t *mif_decode(jas_stream_t *in, const char *optstr);
//JAS_EXPORT
int mif_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
//JAS_EXPORT
int mif_validate(jas_stream_t *in);
#endif

#if defined(JAS_INCLUDE_PNM_CODEC)
/* Format-dependent operations for PNM support. */
//JAS_EXPORT
jas_image_t *pnm_decode(jas_stream_t *in, const char *optstr);
//JAS_EXPORT
int pnm_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
//JAS_EXPORT
int pnm_validate(jas_stream_t *in);
#endif

#if defined(JAS_INCLUDE_RAS_CODEC)
/* Format-dependent operations for Sun Rasterfile support. */
//JAS_EXPORT
jas_image_t *ras_decode(jas_stream_t *in, const char *optstr);
//JAS_EXPORT
int ras_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
//JAS_EXPORT
int ras_validate(jas_stream_t *in);
#endif

#if defined(JAS_INCLUDE_BMP_CODEC)
/* Format-dependent operations for BMP support. */
//JAS_EXPORT
jas_image_t *bmp_decode(jas_stream_t *in, const char *optstr);
//JAS_EXPORT
int bmp_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
//JAS_EXPORT
int bmp_validate(jas_stream_t *in);
#endif

#if defined(JAS_INCLUDE_JP2_CODEC)
/* Format-dependent operations for JP2 support. */
//JAS_EXPORT
jas_image_t *jp2_decode(jas_stream_t *in, const char *optstr);
//JAS_EXPORT
int jp2_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
//JAS_EXPORT
int jp2_validate(jas_stream_t *in);
#endif

#if defined(JAS_INCLUDE_JPC_CODEC)
/* Format-dependent operations for JPEG-2000 code stream support. */
//JAS_EXPORT
jas_image_t *jpc_decode(jas_stream_t *in, const char *optstr);
//JAS_EXPORT
int jpc_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
//JAS_EXPORT
int jpc_validate(jas_stream_t *in);
#endif

#if defined(JAS_INCLUDE_PGX_CODEC)
/* Format-dependent operations for PGX support. */
//JAS_EXPORT
jas_image_t *pgx_decode(jas_stream_t *in, const char *optstr);
//JAS_EXPORT
int pgx_encode(jas_image_t *image, jas_stream_t *out, const char *optstr);
//JAS_EXPORT
int pgx_validate(jas_stream_t *in);
#endif

#ifdef __cplusplus
}
#endif

#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.

```
/*
 * Copyright (c) 2001-2002 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_INIT_H
#define JAS_INIT_H

/* The configuration header file should be included first. */
#include <jasper/jas_config.h>

#include "jasper/jas_malloc.h"
#include "jasper/jas_image.h"
#include "jasper/jas_log.h"

#include <stdarg.h>

#ifdef __cplusplus
extern "C" {
#endif

#if defined(JAS_FOR_INTERNAL_USE_ONLY)
/*
Per-thread library context.
*/
typedef struct {

        /*
        The level of debugging checks/output enabled by the library.
        A larger value corresponds to a greater level of debugging checks/output.
        */
        int debug_level;

         /*
         The function used to output error/warning/informational messages.
         int (*vlogmsgf)(jas_logtype_t type, const char *format, va_list ap);
         */
         jas_vlogmsgf_t *vlogmsgf;

         /*
         The image format information to be used to populate the image format
         table.
         */
         size_t image_numfmts;
         jas_image_fmtinfo_t image_fmtinfos[JAS_IMAGE_MAXFMTS];

         /* The maximum number of samples allowable in an image to be decoded. */
         size_t dec_default_max_samples;

} jas_ctx_t;
#endif

typedef void *jas_context_t;

/*****************************************************************************\
* Library Run-Time Configuration.
\*****************************************************************************/

JAS_EXPORT
void jas_conf_clear(void);
```

```
JAS_EXPORT
void jas_conf_set_multithread(int multithread);

JAS_EXPORT
void jas_conf_set_allocator(jas_allocator_t *allocator);

JAS_EXPORT
void jas_conf_set_debug_level(int debug_level);

JAS_EXPORT
void jas_conf_set_max_mem_usage(size_t max_mem);

JAS_EXPORT
void jas_conf_set_dec_default_max_samples(size_t max_samples);

JAS_EXPORT
void jas_conf_set_vlogmsgf(jas_vlogmsgf_t *func);

/*****************************************************************************\
* Library Initialization and Cleanup.
\*****************************************************************************/

JAS_EXPORT
int jas_init_library(void);

JAS_EXPORT
int jas_cleanup_library(void);

/*****************************************************************************\
* Thread Initialization and Cleanup.
\*****************************************************************************/

JAS_EXPORT
int jas_init_thread(void);

JAS_EXPORT
int jas_cleanup_thread(void);

/*****************************************************************************\
* Legacy Initialization and Cleanup Functions.
\*****************************************************************************/

JAS_EXPORT
int jas_init(void);

JAS_EXPORT
void jas_cleanup(void);

/*****************************************************************************\
* Context Management
\*****************************************************************************/

JAS_EXPORT
jas_context_t jas_context_create(void);

JAS_EXPORT
void jas_context_destroy(jas_context_t context);

JAS_EXPORT
jas_context_t jas_get_default_context(void);

JAS_EXPORT
jas_context_t jas_get_context(void);

JAS_EXPORT
void jas_set_context(jas_context_t context);

/*****************************************************************************\
* Getting/Setting Context Properties
\*****************************************************************************/

/* This function is only for internal use by the library. */
JAS_EXPORT
int jas_get_debug_level_internal(void);

/* This function is only for internal use by the library. */
JAS_EXPORT
size_t jas_get_dec_default_max_samples_internal(void);

/* This function is only for internal use by the library. */
```

```
JAS_EXPORT
jas_vlogmsgf_t *jas_get_vlogmsgf_internal(void);

#if defined(JAS_FOR_INTERNAL_USE_ONLY)
#if defined(JAS_HAVE_THREAD_LOCAL)
extern _Thread_local jas_ctx_t *jas_cur_ctx;
#endif

/* This function is only for internal use by the library. */
jas_ctx_t *jas_get_ctx_internal(void);

/* This function is only for internal use by the library. */
static inline jas_ctx_t *jas_get_ctx(void)
{
#if defined(JAS_HAVE_THREAD_LOCAL)
        return jas_cur_ctx ? jas_cur_ctx : jas_get_ctx_internal();
#else
        return JAS_CAST(jas_ctx_t *, jas_get_ctx_internal());
#endif
}
#endif

JAS_EXPORT
void jas_set_debug_level(int debug_level);

static inline int jas_get_debug_level(void)
{
#if defined(JAS_FOR_INTERNAL_USE_ONLY)
        jas_ctx_t *ctx = jas_get_ctx();
        return ctx->debug_level;
#else
        return jas_get_debug_level_internal();
#endif
}

JAS_EXPORT
void jas_set_dec_default_max_samples(size_t max_samples);

static inline size_t jas_get_dec_default_max_samples(void)
{
#if defined(JAS_FOR_INTERNAL_USE_ONLY)
        jas_ctx_t *ctx = jas_get_ctx();
        return ctx->dec_default_max_samples;
#else
        return jas_get_dec_default_max_samples_internal();
#endif
}

JAS_EXPORT
void jas_set_vlogmsgf(jas_vlogmsgf_t *func);

static inline
jas_vlogmsgf_t *jas_get_vlogmsgf(void)
{
#if defined(JAS_FOR_INTERNAL_USE_ONLY)
        jas_ctx_t *ctx = jas_get_ctx();
        return ctx->vlogmsgf;
#else
        return jas_get_vlogmsgf_internal();
#endif
}

#ifdef __cplusplus
}
#endif

#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.

```
/*
 * Copyright (c) 2001-2002 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_LOG_H
#define JAS_LOG_H

/*****************************************************************************\
* Includes.
\*****************************************************************************/

/* The configuration header file should be included first. */
#include <jasper/jas_config.h>

#include <stdio.h>
#include <stdarg.h>

#ifdef __cplusplus
extern "C" {
```

```
#endif

/*****************************************************************************\
* Macros and functions.
\*****************************************************************************/

#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
#define JAS_LOGTYPE_NUM_CLASSES 5

#define JAS_LOGTYPE_MAX_PRIORITY 16384

// NOTE: without the @struct, jas_logtype_t autolinks are not generated
typedef unsigned int jas_logtype_t;

typedef int (jas_vlogmsgf_t)(jas_logtype_t, const char *, va_list);

static inline jas_logtype_t jas_logtype_init(int clas, int priority)
{
        assert(clas >= 0 && clas < JAS_LOGTYPE_NUM_CLASSES);
        assert(priority >= 0 && priority <= JAS_LOGTYPE_MAX_PRIORITY);
        return (clas & 0xf) | (priority « 4);
}

static inline int jas_logtype_getclass(jas_logtype_t type)
{
        return type & 0xf;
}

static inline int jas_logtype_getpriority(jas_logtype_t type)
{
        return type » 4;
}

JAS_EXPORT
int jas_vlogmsgf(jas_logtype_t type, const char *fmt, va_list ap);

JAS_EXPORT
int jas_vlogmsgf_stderr(jas_logtype_t type, const char *fmt, va_list ap);

JAS_EXPORT
int jas_vlogmsgf_discard(jas_logtype_t type, const char *fmt, va_list ap);

#ifdef __cplusplus
}
#endif

#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.

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */


#ifndef JAS_MALLOC_H
#define JAS_MALLOC_H

/*****************************************************************************\
* Includes.
\*****************************************************************************/

/* The configuration header file should be included first. */
#include <jasper/jas_config.h>

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

```

#include <stdio.h>

#ifdef __cplusplus
extern "C" {
#endif

/*****************************************************************************\
* Types.
\*****************************************************************************/

typedef struct jas_allocator_s {

        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);

        void (*(reserved[4]))(void);

} jas_allocator_t;

typedef struct {

        /* The base class. */
        jas_allocator_t base;

} jas_std_allocator_t;

#if defined(JAS_FOR_INTERNAL_USE_ONLY)
/*
The allocator wrapper type.
This type is an allocator that adds memory usage tracking to another
allocator.
The allocator wrapper does not directly perform memory allocation itself.
Instead, it delegate to another allocator.
*/
typedef struct {

        /* The base class. */
        jas_allocator_t base;

        /* The delegated-to allocator. */
        jas_allocator_t *delegate;

        /* The maximum amount of memory that can be used by the allocator. */
        size_t max_mem;

        /* The current amount of memory in use by the allocator. */
        size_t mem;

#if defined(JAS_THREADS)
        /* A mutex for synchronized access to the allocator. */
        jas_mutex_t mutex;
#endif

} jas_basic_allocator_t;
#endif

/*****************************************************************************\
* Data.
\*****************************************************************************/

#if defined(JAS_FOR_INTERNAL_USE_ONLY)
extern jas_allocator_t *jas_allocator;
extern jas_std_allocator_t jas_std_allocator;
extern jas_basic_allocator_t jas_basic_allocator;
#endif

/*****************************************************************************\
* Functions.
\*****************************************************************************/

JAS_EXPORT
void *jas_malloc(size_t size);

JAS_EXPORT
```

```
void jas_free(void *ptr);

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

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

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

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

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

JAS_EXPORT
void jas_set_max_mem_usage(size_t max_mem);

JAS_EXPORT
size_t jas_get_mem_usage(void);

JAS_EXPORT
void jas_std_allocator_init(jas_std_allocator_t *allocator);

JAS_EXPORT
void jas_allocator_cleanup(jas_allocator_t *allocator);

#if defined(JAS_FOR_INTERNAL_USE_ONLY)

/* This function is for internal library use only. */
void jas_set_allocator(jas_allocator_t* allocator);

/* This function is for internal library use only. */
void jas_basic_allocator_init(jas_basic_allocator_t *allocator,
  jas_allocator_t *delegate, size_t max_mem);

#endif

JAS_EXPORT
size_t jas_get_total_mem_size(void);

#ifdef __cplusplus
}
#endif

#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.

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_MATH_H
#define JAS_MATH_H

/*****************************************************************************\
* Includes
\*****************************************************************************/

/* The configuration header file should be included first. */
#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>

#ifdef __cplusplus
extern "C" {
#endif

/*****************************************************************************\
* Macros
\*****************************************************************************/

#define JAS_KIBI        JAS_CAST(size_t, 1024)
#define JAS_MEBI        (JAS_KIBI * JAS_KIBI)

/* Compute the absolute value. */
#define JAS_ABS(x) \
        (((x) >= 0) ? (x) : (-(x)))

/* Compute the minimum of two values. */
#define JAS_MIN(x, y) \
        (((x) < (y)) ? (x) : (y))

/* Compute the maximum of two values. */
#define JAS_MAX(x, y) \
        (((x) > (y)) ? (x) : (y))

/* Compute the remainder from division (where division is defined such
  that the remainder is always nonnegative). */
#define JAS_MOD(x, y) \
        (((x) < 0) ? (((-x) % (y)) ? ((y) - ((-(x)) % (y))) : (0)) : ((x) % (y)))

/* Compute the integer with the specified number of least significant bits
  set to one. */
#define JAS_ONES(n) \
  ((1 « (n)) - 1)

/*****************************************************************************\
*
\*****************************************************************************/

#if defined(__clang__) || (defined(__GNUC__) && __GNUC__ > 6)
/* suppress clang warning "shifting a negative signed value is
   undefined" in the assertions below */
#pragma GCC diagnostic push
#pragma GCC diagnostic ignored "-Wshift-negative-value"
#endif

JAS_ATTRIBUTE_CONST
JAS_ATTRIBUTE_DISABLE_UBSAN
inline static int jas_int_asr(int x, unsigned n)
{
        // Ensure that the shift of a negative value appears to behave as a
        // signed arithmetic shift.
        assert(((-1) » 1) == -1);
        // The behavior is undefined when x is negative.
        // We tacitly assume the behavior is equivalent to a signed
        // arithmetic right shift.
        return x » n;
}

JAS_ATTRIBUTE_CONST
JAS_ATTRIBUTE_DISABLE_UBSAN
inline static int jas_int_asl(int x, unsigned n)
{
        // Ensure that the shift of a negative value appears to behave as a
        // signed arithmetic shift.
        assert(((-1) « 1) == -2);
        // The behavior is undefined when x is negative.
        // We tacitly assume the behavior is equivalent to a signed
        // arithmetic left shift.
        return x « n;
}

JAS_ATTRIBUTE_CONST
JAS_ATTRIBUTE_DISABLE_UBSAN
inline static int_least32_t jas_least32_asr(int_least32_t x, unsigned n)
{
        // Ensure that the shift of a negative value appears to behave as a
        // signed arithmetic shift.
```

```
        assert(((JAS_CAST(int_least32_t, -1)) » 1) == JAS_CAST(int_least32_t, -1));
        // The behavior is undefined when x is negative.
        // We tacitly assume the behavior is equivalent to a signed
        // arithmetic right shift.
        return x » n;
}

JAS_ATTRIBUTE_CONST
JAS_ATTRIBUTE_DISABLE_UBSAN
inline static int_least32_t jas_least32_asl(int_least32_t x, unsigned n)
{
        // Ensure that the shift of a negative value appears to behave as a
        // signed arithmetic shift.
        assert(((JAS_CAST(int_least32_t, -1)) « 1) == JAS_CAST(int_least32_t, -2));
        // The behavior is undefined when x is negative.
        // We tacitly assume the behavior is equivalent to a signed
        // arithmetic left shift.
        return x « n;
}

JAS_ATTRIBUTE_CONST
JAS_ATTRIBUTE_DISABLE_UBSAN
inline static int_fast32_t jas_fast32_asr(int_fast32_t x, unsigned n)
{
        // Ensure that the shift of a negative value appears to behave as a
        // signed arithmetic shift.
        assert(((JAS_CAST(int_fast32_t, -1)) » 1) == JAS_CAST(int_fast32_t, -1));
        // The behavior is undefined when x is negative.
        // We tacitly assume the behavior is equivalent to a signed
        // arithmetic right shift.
        return x » n;
}

JAS_ATTRIBUTE_CONST
JAS_ATTRIBUTE_DISABLE_UBSAN
inline static int_fast32_t jas_fast32_asl(int_fast32_t x, unsigned n)
{
        // Ensure that the shift of a negative value appears to behave as a
        // signed arithmetic shift.
        assert(((JAS_CAST(int_fast32_t, -1)) « 1) == JAS_CAST(int_fast32_t, -2));
        // The behavior is undefined when x is negative.
        // We tacitly assume the behavior is equivalent to a signed
        // arithmetic left shift.
        return x « n;
}

#if defined(__clang__) || (defined(__GNUC__) && __GNUC__ > 6)
#pragma GCC diagnostic pop
#endif

/*****************************************************************************\
* Safe integer arithmetic (i.e., with overflow checking).
\*****************************************************************************/

/* Compute the product of two size_t integers with overflow checking. */
inline static bool jas_safe_size_mul(size_t x, size_t y, size_t *result)
{
#if jas_has_builtin(__builtin_mul_overflow) || (defined(__GNUC__) && __GNUC__ > 5)
        size_t result_buffer;
        bool valid = !__builtin_mul_overflow(x, y, &result_buffer);
        if (valid && result) {
                *result = result_buffer;
        }
        return valid;
#else
        /* Check if overflow would occur */
        if (x && y > SIZE_MAX / x) {
                /* Overflow would occur. */
                return false;
        }
        if (result) {
                *result = x * y;
        }
        return true;
#endif
}

/* Compute the product of three size_t integers with overflow checking. */
inline static bool jas_safe_size_mul3(size_t a, size_t b, size_t c,
  size_t *result)
{
```

```
        size_t tmp;
        if (!jas_safe_size_mul(a, b, &tmp) ||
          !jas_safe_size_mul(tmp, c, &tmp)) {
                return false;
        }
        if (result) {
                *result = tmp;
        }
        return true;
}

/* Compute the sum of two size_t integers with overflow checking. */
inline static bool jas_safe_size_add(size_t x, size_t y, size_t *result)
{
#if jas_has_builtin(__builtin_add_overflow) || (defined(__GNUC__) && __GNUC__ > 5)
        size_t result_buffer;
        bool valid = !__builtin_add_overflow(x, y, &result_buffer);
        if (valid && result) {
                *result = result_buffer;
        }
        return valid;
#else
        if (y > SIZE_MAX - x) {
                return false;
        }
        if (result) {
                *result = x + y;
        }
        return true;
#endif
}

/* Compute the difference of two size_t integers with overflow checking. */
inline static bool jas_safe_size_sub(size_t x, size_t y, size_t *result)
{
#if jas_has_builtin(__builtin_sub_overflow) || (defined(__GNUC__) && __GNUC__ > 5)
        size_t result_buffer;
        bool valid = !__builtin_sub_overflow(x, y, &result_buffer);
        if (valid && result) {
                *result = result_buffer;
        }
        return valid;
#else
        if (y > x) {
                return false;
        }
        if (result) {
                *result = x - y;
        }
        return true;
#endif
}

/* Compute the product of two int_fast32_t integers with overflow checking. */
inline static bool jas_safe_intfast32_mul(int_fast32_t x, int_fast32_t y,
  int_fast32_t *result)
{
#if jas_has_builtin(__builtin_mul_overflow) || (defined(__GNUC__) && __GNUC__ > 5)
        int_fast32_t result_buffer;
        bool valid = !__builtin_mul_overflow(x, y, &result_buffer);
        if (valid && result) {
                *result = result_buffer;
        }
        return valid;
#else
        if (x > 0) {
                /* x is positive */
                if (y > 0) {
                        /* x and y are positive */
                        if (x > INT_FAST32_MAX / y) {
                                return false;
                        }
                } else {
                        /* x positive, y nonpositive */
                        if (y < INT_FAST32_MIN / x) {
                                return false;
                        }
                }
        } else {
                /* x is nonpositive */
                if (y > 0) {
```

```
                        /* x is nonpositive, y is positive */
                        if (x < INT_FAST32_MIN / y) {
                                return false;
                        }
                } else { /* x and y are nonpositive */
                        if (x != 0 && y < INT_FAST32_MAX / x) {
                                return false;
                        }
                }
        }

        if (result) {
                *result = x * y;
        }
        return true;
#endif
}

/* Compute the product of three int_fast32_t integers with overflow checking. */
inline static bool jas_safe_intfast32_mul3(int_fast32_t a, int_fast32_t b,
  int_fast32_t c, int_fast32_t *result)
{
        int_fast32_t tmp;
        if (!jas_safe_intfast32_mul(a, b, &tmp) ||
          !jas_safe_intfast32_mul(tmp, c, &tmp)) {
                return false;
        }
        if (result) {
                *result = tmp;
        }
        return true;
}

/* Compute the sum of two int_fast32_t integers with overflow checking. */
inline static bool jas_safe_intfast32_add(int_fast32_t x, int_fast32_t y,
  int_fast32_t *result)
{
#if jas_has_builtin(__builtin_add_overflow) || (defined(__GNUC__) && __GNUC__ > 5)
        int_fast32_t result_buffer;
        bool valid = !__builtin_add_overflow(x, y, &result_buffer);
        if (valid && result) {
                *result = result_buffer;
        }
        return valid;
#else
        if ((y > 0 && x > INT_FAST32_MAX - y) ||
          (y < 0 && x < INT_FAST32_MIN - y)) {
                return false;
        }
        if (result) {
                *result = x + y;
        }
        return true;
#endif
}

#if 0
/*
This function is potentially useful but not currently used.
So, it is commented out.
*/
inline static bool jas_safe_uint_mul(unsigned x, unsigned y, unsigned *result)
{
        /* Check if overflow would occur */
        if (x && y > UINT_MAX / x) {
                /* Overflow would occur. */
                return false;
        }
        if (result) {
                *result = x * y;
        }
        return true;
}
#endif

/*****************************************************************************\
* Safe integer arithmetic (i.e., with overflow checking).
\*****************************************************************************/

typedef struct {
        bool valid;
```

```
        uint_least64_t value;
} jas_safeui64_t;

static inline
jas_safeui64_t jas_safeui64_from_intmax(intmax_t x)
{
        jas_safeui64_t result;
        if (x >= 0 && x <= UINT_LEAST64_MAX) {
                result.valid = true;
                result.value = JAS_CAST(uint_least64_t, x);
        } else {
                result.valid = false;
                result.value = 0;
        }
        return result;
}

static inline
jas_safeui64_t jas_safeui64_add(jas_safeui64_t x, jas_safeui64_t y)
{
        jas_safeui64_t result;
        if (x.valid && y.valid && y.value <= UINT_LEAST64_MAX - x.value) {
                result.valid = true;
                result.value = x.value + y.value;
        } else {
                result.valid = false;
                result.value = 0;
        }
        return result;
}

static inline
jas_safeui64_t jas_safeui64_sub(jas_safeui64_t x, jas_safeui64_t y)
{
        jas_safeui64_t result;
        if (x.valid && y.valid && y.value <= x.value) {
                result.valid = true;
                result.value = x.value - y.value;
        } else {
                result.valid = false;
                result.value = 0;
        }
        return result;
}

static inline
jas_safeui64_t jas_safeui64_mul(jas_safeui64_t x, jas_safeui64_t y)
{
        jas_safeui64_t result;
        if (!x.valid || !y.valid || (x.value && y.value > UINT_LEAST64_MAX /
          x.value)) {
                result.valid = false;
                result.value = 0;
        } else {
                result.valid = true;
                result.value = x.value * y.value;
        }
        return result;
}

static inline
jas_safeui64_t jas_safeui64_div(jas_safeui64_t x, jas_safeui64_t y)
{
        jas_safeui64_t result;
        if (x.valid && y.valid && y.value) {
                result.valid = true;
                result.value = x.value / y.valid;
        } else {
                result.valid = false;
                result.value = 0;
        }
        return result;
}

static inline
jas_safeui64_t jas_safeui64_pow2_intmax(intmax_t x)
{
        jas_safeui64_t result;
        if (x >= 0 && x < 64) {
                result.valid = true;
                result.value = JAS_CAST(uint_least64_t, 1) « x;
```

```
        } else {
                result.valid = false;
                result.value = 0;
        }
        return result;
}

static inline
int jas_safeui64_to_int(jas_safeui64_t x, int invalid_value)
{
        int result;
        if (x.valid && x.value <= INT_MAX) {
                result = JAS_CAST(int, x.value);
        } else {
                result = invalid_value;
        }
        return result;
}

/*****************************************************************************\
\*****************************************************************************/

#ifdef __cplusplus
}
#endif

#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.

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_SEQ_H
#define JAS_SEQ_H

```

```
/*****************************************************************************\
* Includes.
\*****************************************************************************/

/* The configuration header file should be included first. */
#include <jasper/jas_config.h> /* IWYU pragma: keep */

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

#include <stdio.h>

#ifdef __cplusplus
extern "C" {
#endif

/*****************************************************************************\
* Constants.
\*****************************************************************************/

/* This matrix is a reference to another matrix. */
#define JAS_MATRIX_REF  0x0001

/*****************************************************************************\
* Types.
\*****************************************************************************/

/* An element in a sequence. */
#ifdef JAS_ENABLE_32BIT
typedef int_least32_t jas_seqent_t;
#define PRIjas_seqent PRIiLEAST32
#else
typedef int_fast32_t jas_seqent_t;
#define PRIjas_seqent PRIiFAST32
#endif

/* An element in a matrix. */
#ifdef JAS_ENABLE_32BIT
typedef int_least32_t jas_matent_t;
#else
typedef int_fast32_t jas_matent_t;
#endif

#ifdef JAS_ENABLE_32BIT
typedef int_least32_t jas_matind_t;
#else
typedef int_fast32_t jas_matind_t;
#endif

typedef struct {

        /* Additional state information. */
        int flags_;

        /* The starting horizontal index. */
        jas_matind_t xstart_;

        /* The starting vertical index. */
        jas_matind_t ystart_;

        /* The ending horizontal index. */
        jas_matind_t xend_;

        /* The ending vertical index. */
        jas_matind_t yend_;

        /* The number of rows in the matrix. */
        jas_matind_t numrows_;

        /* The number of columns in the matrix. */
        jas_matind_t numcols_;

        /* Pointers to the start of each row. */
        jas_seqent_t **rows_;

        /* The allocated size of the rows array. */
        int_fast32_t maxrows_;

        /* The matrix data buffer. */
        jas_seqent_t *data_;

```

```
        /* The allocated size of the data array. */
        int_fast32_t datasize_;

} jas_matrix_t;

typedef jas_matrix_t jas_seq2d_t;

typedef jas_matrix_t jas_seq_t;

/*****************************************************************************\
* Functions/macros for matrix class.
\*****************************************************************************/

JAS_ATTRIBUTE_PURE
static inline jas_matind_t jas_matrix_numrows(const jas_matrix_t *matrix)
{
        return matrix->numrows_;
}

JAS_ATTRIBUTE_PURE
static inline jas_matind_t jas_matrix_numcols(const jas_matrix_t *matrix)
{
        return matrix->numcols_;
}

JAS_ATTRIBUTE_PURE
static inline jas_matind_t jas_matrix_size(const jas_matrix_t *matrix)
{
        return jas_matrix_numcols(matrix) * jas_matrix_numrows(matrix);
}

JAS_ATTRIBUTE_PURE
static inline bool jas_matrix_empty(const jas_matrix_t *matrix)
{
        return jas_matrix_numcols(matrix) == 0 || jas_matrix_numrows(matrix) == 0;
}

JAS_ATTRIBUTE_PURE
static inline jas_seqent_t jas_matrix_get(const jas_matrix_t *matrix, jas_matind_t i, jas_matind_t j)
{
        assert(i >= 0 && i < matrix->numrows_ && j >= 0 && j < matrix->numcols_);
        return matrix->rows_[i][j];
}

static inline void jas_matrix_set(jas_matrix_t *matrix, jas_matind_t i, jas_matind_t j, jas_seqent_t v)
{
        assert(i >= 0 && i < matrix->numrows_ && j >= 0 && j < matrix->numcols_);
        matrix->rows_[i][j] = v;
}

JAS_ATTRIBUTE_PURE
static inline jas_seqent_t jas_matrix_getv(const jas_matrix_t *matrix, jas_matind_t i)
{
        return matrix->numrows_ == 1
                ? matrix->rows_[0][i]
                : matrix->rows_[i][0];
}

static inline void jas_matrix_setv(jas_matrix_t *matrix, jas_matind_t i, jas_seqent_t v)
{
        if (matrix->numrows_ == 1)
                matrix->rows_[0][i] = v;
        else
                matrix->rows_[i][0] = v;
}

JAS_ATTRIBUTE_PURE
static inline jas_seqent_t *jas_matrix_getref(const jas_matrix_t *matrix, jas_matind_t i, jas_matind_t j)
{
        return &matrix->rows_[i][j];
}

JAS_ATTRIBUTE_PURE
static inline jas_seqent_t *jas_matrix_getvref(const jas_matrix_t *matrix, jas_matind_t i)
{
        return matrix->numrows_ > 1
                ? jas_matrix_getref(matrix, i, 0)
                : jas_matrix_getref(matrix, 0, i);
}

JAS_EXPORT
```

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

JAS_EXPORT
void jas_matrix_destroy(jas_matrix_t *matrix);

JAS_EXPORT
int jas_matrix_resize(jas_matrix_t *matrix, jas_matind_t numrows, jas_matind_t numcols);

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

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);

static inline int jas_matrix_bindrow(jas_matrix_t *mat0, jas_matrix_t *mat1, jas_matind_t r)
{
        return jas_matrix_bindsub(mat0, mat1, r, 0, r, mat1->numcols_ - 1);
}

static inline int jas_matrix_bindcol(jas_matrix_t *mat0, jas_matrix_t *mat1, jas_matind_t c)
{
        return jas_matrix_bindsub(mat0, mat1, 0, c, mat1->numrows_ - 1, c);
}

JAS_EXPORT
void jas_matrix_clip(jas_matrix_t *matrix, jas_seqent_t minval,
  jas_seqent_t maxval);

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

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

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

JAS_EXPORT
void jas_matrix_setall(jas_matrix_t *matrix, jas_seqent_t val);

JAS_ATTRIBUTE_PURE
static inline size_t jas_matrix_rowstep(const jas_matrix_t *matrix)
{
        return matrix->numrows_ > 1
                ? (size_t)(matrix->rows_[1] - matrix->rows_[0])
                : 0u;
}

JAS_ATTRIBUTE_PURE
static inline size_t jas_matrix_step(const jas_matrix_t *matrix)
{
        return matrix->numrows_ > 1
                ? jas_matrix_rowstep(matrix)
                : 1;
}

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

JAS_EXPORT
jas_matrix_t *jas_matrix_copy(jas_matrix_t *x);

JAS_EXPORT
jas_matrix_t *jas_matrix_input(FILE *);

JAS_ATTRIBUTE_CONST
static inline jas_seqent_t jas_seqent_asl(jas_seqent_t x, unsigned n)
{
#ifdef JAS_ENABLE_32BIT
        return jas_least32_asl(x, n);
#else
        return jas_fast32_asl(x, n);
#endif
}

JAS_ATTRIBUTE_CONST
static inline jas_seqent_t jas_seqent_asr(jas_seqent_t x, unsigned n)
{
#ifdef JAS_ENABLE_32BIT
        return jas_least32_asr(x, n);
```

```
#else
        return jas_fast32_asr(x, n);
#endif
}

/*****************************************************************************\
* Functions/macros for 2-D sequence class.
\*****************************************************************************/

JAS_EXPORT
jas_seq2d_t *jas_seq2d_copy(jas_seq2d_t *x);

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

static inline void jas_seq2d_destroy(jas_seq2d_t *s)
{
        jas_matrix_destroy(s);
}

JAS_ATTRIBUTE_PURE
static inline jas_matind_t jas_seq2d_xstart(const jas_seq2d_t *s)
{
        return s->xstart_;
}

JAS_ATTRIBUTE_PURE
static inline jas_matind_t jas_seq2d_ystart(const jas_seq2d_t *s)
{
        return s->ystart_;
}

JAS_ATTRIBUTE_PURE
static inline jas_matind_t jas_seq2d_xend(const jas_seq2d_t *s)
{
        return s->xend_;
}

JAS_ATTRIBUTE_PURE
static inline jas_matind_t jas_seq2d_yend(const jas_seq2d_t *s)
{
        return s->yend_;
}

JAS_ATTRIBUTE_PURE
static inline jas_seqent_t *jas_seq2d_getref(const jas_seq2d_t *s, jas_matind_t x, jas_matind_t y)
{
        return jas_matrix_getref(s, y - s->ystart_, x - s->xstart_);
}

JAS_ATTRIBUTE_PURE
static inline jas_seqent_t jas_seq2d_get(const jas_seq2d_t *s, jas_matind_t x, jas_matind_t y)
{
        return jas_matrix_get(s, y - s->ystart_, x - s->xstart_);
}

JAS_ATTRIBUTE_PURE
static inline size_t jas_seq2d_rowstep(const jas_seq2d_t *s)
{
        return jas_matrix_rowstep(s);
}

JAS_ATTRIBUTE_PURE
static inline unsigned jas_seq2d_width(const jas_seq2d_t *s)
{
        return (unsigned)(s->xend_ - s->xstart_);
}

JAS_ATTRIBUTE_PURE
static inline unsigned jas_seq2d_height(const jas_seq2d_t *s)
{
        return (unsigned)(s->yend_ - s->ystart_);
}

static inline void jas_seq2d_setshift(jas_seq2d_t *s, jas_matind_t x, jas_matind_t y)
{
        s->xstart_ = x;
        s->ystart_ = y;
        s->xend_ = s->xstart_ + s->numcols_;
        s->yend_ = s->ystart_ + s->numrows_;
```

```
}

JAS_ATTRIBUTE_PURE
static inline jas_matind_t jas_seq2d_size(const jas_seq2d_t *s)
{
        return jas_seq2d_width(s) * jas_seq2d_height(s);
}

JAS_ATTRIBUTE_PURE
static inline bool jas_seq2d_empty(const jas_seq2d_t *s)
{
        return jas_seq2d_width(s) == 0 || jas_seq2d_height(s) == 0;
}

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);

/*****************************************************************************\
* Functions/macros for 1-D sequence class.
\*****************************************************************************/

static inline jas_seq_t *jas_seq_create(jas_matind_t start, jas_matind_t end)
{
        return jas_seq2d_create(start, 0, end, 1);
}

static inline void jas_seq_destroy(jas_seq_t *seq)
{
        jas_seq2d_destroy(seq);
}

static inline void jas_seq_set(jas_seq_t *seq, jas_matind_t i, jas_seqent_t v)
{
        seq->rows_[0][i - seq->xstart_] = v;
}

JAS_ATTRIBUTE_PURE
static inline jas_seqent_t *jas_seq_getref(const jas_seq_t *seq, jas_matind_t i)
{
        return &seq->rows_[0][i - seq->xstart_];
}

JAS_ATTRIBUTE_PURE
static inline jas_seqent_t jas_seq_get(const jas_seq_t *seq, jas_matind_t i)
{
        return seq->rows_[0][i - seq->xstart_];
}

JAS_ATTRIBUTE_PURE
static inline jas_matind_t jas_seq_start(const jas_seq_t *seq)
{
        return seq->xstart_;
}

JAS_ATTRIBUTE_PURE
static inline jas_matind_t jas_seq_end(const jas_seq_t *seq)
{
        return seq->xend_;
}

#ifdef __cplusplus
}
#endif

#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_getrwlimit(stream) (((const jas_stream_t *)(stream))->rwlimit_)

  Get the read/write limit for a stream.
- #define jas_stream_getrwcount(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_setrwlimit (jas_stream_t *stream, long rwlimit)

  Set the read/write limit for a stream.
- JAS_EXPORT long jas_stream_setrwcount (jas_stream_t *stream, long rw_count)

  Set the read/write count for a stream.
- JAS_EXPORT size_t jas_stream_read (jas_stream_t *stream, void *buffer, size_t count)

  Read characters from a stream into a buffer.

- JAS_EXPORT unsigned jas_stream_peek (jas_stream_t *stream, void *buffer, size_t count)

  Attempt to retrieve one or more pending characters of input from a stream into a buffer without actually removing the characters from the stream.
- JAS_EXPORT size_t jas_stream_write (jas_stream_t *stream, const void *buffer, size_t count)

  Write characters from a buffer to a stream.
- JAS_EXPORT int jas_stream_printf (jas_stream_t *stream, const char *format,...)

  Write formatted output to a stream.
- JAS_EXPORT int jas_stream_puts (jas_stream_t *stream, const char *s)

  Write a string to a stream.
- JAS_EXPORT char * jas_stream_gets (jas_stream_t *stream, char *buffer, int buffer_size)

  Read a line of input from a stream.
- JAS_EXPORT int jas_stream_ungetc (jas_stream_t *stream, int c)

  Put a character back on a stream.
- JAS_EXPORT JAS_ATTRIBUTE_PURE int jas_stream_isseekable (jas_stream_t *stream)

  Determine if stream supports seeking.
- JAS_EXPORT long jas_stream_seek (jas_stream_t *stream, long offset, int origin)

  Set the current position within the stream.
- JAS_EXPORT long jas_stream_tell (jas_stream_t *stream)

  Get the current position within the stream.
- JAS_EXPORT int jas_stream_rewind (jas_stream_t *stream)

  Seek to the beginning of a stream.
- JAS_EXPORT int jas_stream_flush (jas_stream_t *stream)

  Flush any pending output to a stream.
- JAS_EXPORT int jas_stream_copy (jas_stream_t *destination, jas_stream_t *source, ssize_t count)

  Copy data from one stream to another.
- JAS_EXPORT int jas_stream_display (jas_stream_t *stream, FILE *fp, int count)

  Print a hex dump of data read from a stream.
- JAS_EXPORT ssize_t jas_stream_gobble (jas_stream_t *stream, size_t count)

  Consume (i.e., discard) characters from stream.
- JAS_EXPORT ssize_t jas_stream_pad (jas_stream_t *stream, size_t count, int value)

  Write a fill character multiple times to a stream.
- JAS_EXPORT long jas_stream_length (jas_stream_t *stream)

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

16.29.1 Detailed Description

I/O Stream Class.

16.30 jas_stream.h

Go to the documentation of this file.

```
/*
 * Copyright (c) 1999-2000 Image Power, Inc. and the University of
 *   British Columbia.
 * Copyright (c) 2001-2003 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_STREAM_H
#define JAS_STREAM_H

/*****************************************************************************\
* Includes.
\*****************************************************************************/

/* The configuration header file should be included first. */
#include <jasper/jas_config.h> /* IWYU pragma: export */

#include <stdio.h>
#if defined(JAS_HAVE_FCNTL_H)
#include <fcntl.h>
```

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

#ifdef __cplusplus
extern "C" {
#endif

/******************************************************************************\
* Constants.
\******************************************************************************/

/* On most UNIX systems, we probably need to define O_BINARY ourselves. */
#ifndef O_BINARY
#define O_BINARY        0
#endif

/*
 * Stream open flags.
 */

/* The stream was opened for reading. */
#define JAS_STREAM_READ 0x0001
/* The stream was opened for writing. */
#define JAS_STREAM_WRITE        0x0002
/* The stream was opened for appending. */
#define JAS_STREAM_APPEND       0x0004
/* The stream was opened in binary mode. */
#define JAS_STREAM_BINARY       0x0008
/* The stream should be created/truncated. */
#define JAS_STREAM_CREATE       0x0010

/*
 * Stream buffering flags.
 */

/* The stream is unbuffered. */
#define JAS_STREAM_UNBUF        0x0000
/* The stream is line buffered. */
#define JAS_STREAM_LINEBUF      0x0001
/* The stream is fully buffered. */
#define JAS_STREAM_FULLBUF      0x0002
/* The buffering mode mask. */
#define JAS_STREAM_BUFMODEMASK  0x000f

/* The memory associated with the buffer needs to be deallocated when the
  stream is destroyed. */
#define JAS_STREAM_FREEBUF      0x0008
/* The buffer is currently being used for reading. */
#define JAS_STREAM_RDBUF        0x0010
/* The buffer is currently being used for writing. */
#define JAS_STREAM_WRBUF        0x0020

/*
 * Stream error flags.
 */

/* The end-of-file has been encountered (on reading). */
#define JAS_STREAM_EOF  0x0001
/* An I/O error has been encountered on the stream. */
#define JAS_STREAM_ERR  0x0002
/* The read/write limit has been exceeded. */
#define JAS_STREAM_RWLIMIT      0x0004
/* The error mask. */
#define JAS_STREAM_ERRMASK \
        (JAS_STREAM_EOF | JAS_STREAM_ERR | JAS_STREAM_RWLIMIT)

/*
 * Other miscellaneous constants.
 */

/* The default buffer size (for fully-buffered operation). */
#define JAS_STREAM_BUFSIZE      8192
/* The default permission mask for file creation. */
#define JAS_STREAM_PERMS        0666

/* The maximum number of characters that can always be put back on a stream. */
#define JAS_STREAM_MAXPUTBACK   16

/******************************************************************************\
* Types.
\******************************************************************************/
```

```
/*
 * Generic file object.
 */

typedef void jas_stream_obj_t;

/*
 * Generic file object operations.
 */

typedef struct {

        /* Read characters from a file object. */
        ssize_t (*read_)(jas_stream_obj_t *obj, char *buf, size_t cnt);

        /* Write characters to a file object. */
        ssize_t (*write_)(jas_stream_obj_t *obj, const char *buf, size_t cnt);

        /* Set the position for a file object. */
        long (*seek_)(jas_stream_obj_t *obj, long offset, int origin);

        /* Close a file object. */
        int (*close_)(jas_stream_obj_t *obj);

} jas_stream_ops_t;

typedef struct {

        /* The mode in which the stream was opened. */
        int openmode_;

        /* The buffering mode. */
        int bufmode_;

        /* The stream status. */
        int flags_;

        /* The start of the buffer area to use for reading/writing. */
        jas_uchar *bufbase_;

        /* The start of the buffer area excluding the extra initial space for
          character putback. */
        jas_uchar *bufstart_;

        /* The buffer size. */
        int bufsize_;

        /* The current position in the buffer. */
        jas_uchar *ptr_;

        /* The number of characters that must be read/written before
        the buffer needs to be filled/flushed. */
        int cnt_;

        /* A trivial buffer to be used for unbuffered operation. */
        jas_uchar tinybuf_[JAS_STREAM_MAXPUTBACK + 1];

        /* The operations for the underlying stream file object. */
        const jas_stream_ops_t *ops_;

        /* The underlying stream file object. */
        jas_stream_obj_t *obj_;

        /* The number of characters read/written. */
        long rwcnt_;

        /* The maximum number of characters that may be read/written. */
        long rwlimit_;

} jas_stream_t;

/*
 * Regular file object.
 */

/*
 * File descriptor file object.
 */
typedef struct {
        int fd;
```

```
        int flags;
        char pathname[L_tmpnam + 1];
} jas_stream_fileobj_t;

/* Delete underlying file object upon stream close. */
#define JAS_STREAM_FILEOBJ_DELONCLOSE   0x01
/* Do not close underlying file object upon stream close. */
#define JAS_STREAM_FILEOBJ_NOCLOSE      0x02

/*
 * Memory file object.
 */

typedef struct {

        /* The data associated with this file. */
        jas_uchar *buf_;

        /* The allocated size of the buffer for holding file data. */
        size_t bufsize_;

        /* The length of the file. */
        size_t len_;

        /* The seek position. */
        size_t pos_;

        /* Is the buffer growable? */
        int growable_;

        /* Was the buffer allocated internally? */
        int myalloc_;

} jas_stream_memobj_t;

/*****************************************************************************\
* Macros/functions for opening and closing streams.
\*****************************************************************************/

JAS_EXPORT
jas_stream_t *jas_stream_fopen(const char *filename, const char *mode);

JAS_EXPORT
jas_stream_t *jas_stream_memopen(char *buffer, size_t buffer_size);

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

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

JAS_EXPORT
jas_stream_t *jas_stream_freopen(const char *path, const char *mode, FILE *fp);

JAS_EXPORT
jas_stream_t *jas_stream_tmpfile(void);

JAS_EXPORT
int jas_stream_close(jas_stream_t *stream);

/*****************************************************************************\
* Macros/functions for getting/setting the stream state.
\*****************************************************************************/

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

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

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

#define jas_stream_getrwlimit(stream) \
        (((const jas_stream_t *)(stream))->rwlimit_)

JAS_EXPORT long jas_stream_setrwlimit(jas_stream_t *stream, long rwlimit);

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

```
JAS_EXPORT
long jas_stream_setrwcount(jas_stream_t *stream, long rw_count);

/*****************************************************************************\
* Macros/functions for I/O.
\*****************************************************************************/

/* Read a character from a stream. */
#ifndef NDEBUG
#define jas_stream_getc(stream) jas_stream_getc_func(stream)
#else
#define jas_stream_getc(stream) jas_stream_getc_macro(stream)
#endif

/* Write a character to a stream. */
#ifndef NDEBUG
#define jas_stream_putc(stream, c)      jas_stream_putc_func(stream, c)
#else
#define jas_stream_putc(stream, c)      jas_stream_putc_macro(stream, c)
#endif

JAS_EXPORT
size_t jas_stream_read(jas_stream_t *stream, void *buffer, size_t count);

JAS_EXPORT
unsigned jas_stream_peek(jas_stream_t *stream, void *buffer, size_t count);

JAS_EXPORT
size_t jas_stream_write(jas_stream_t *stream, const void *buffer,
  size_t count);

JAS_EXPORT
int jas_stream_printf(jas_stream_t *stream, const char *format, ...);

JAS_EXPORT
int jas_stream_puts(jas_stream_t *stream, const char *s);

JAS_EXPORT
char *jas_stream_gets(jas_stream_t *stream, char *buffer, int buffer_size);

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

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

/*****************************************************************************\
* Macros/functions for getting/setting the stream position.
\*****************************************************************************/

JAS_EXPORT
JAS_ATTRIBUTE_PURE
int jas_stream_isseekable(jas_stream_t *stream);

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

JAS_EXPORT
long jas_stream_tell(jas_stream_t *stream);

JAS_EXPORT
int jas_stream_rewind(jas_stream_t *stream);

/*****************************************************************************\
* Macros/functions for flushing.
\*****************************************************************************/

JAS_EXPORT
int jas_stream_flush(jas_stream_t *stream);

/*****************************************************************************\
* Miscellaneous macros/functions.
\*****************************************************************************/

JAS_EXPORT
int jas_stream_copy(jas_stream_t *destination, jas_stream_t *source,
  ssize_t count);

JAS_EXPORT
```

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

JAS_EXPORT
ssize_t jas_stream_gobble(jas_stream_t *stream, size_t count);

JAS_EXPORT
ssize_t jas_stream_pad(jas_stream_t *stream, size_t count, int value);

JAS_EXPORT
long jas_stream_length(jas_stream_t *stream);

/*****************************************************************************\
* Internal functions.
\*****************************************************************************/

/* The following functions are for internal use only!  If you call them
directly, you will die a horrible, miserable, and painful death! */

/* These prototypes need to be here for the sake of the stream_getc and
stream_putc macros. */
/* Library users must not invoke these functions directly. */
JAS_EXPORT int jas_stream_fillbuf(jas_stream_t *stream, int getflag);
JAS_EXPORT int jas_stream_flushbuf(jas_stream_t *stream, int c);
JAS_EXPORT int jas_stream_getc_func(jas_stream_t *stream);
JAS_EXPORT int jas_stream_putc_func(jas_stream_t *stream, int c);

/* Read a character from a stream. */
static inline int jas_stream_getc2(jas_stream_t *stream)
{
        if (--stream->cnt_ < 0)
                return jas_stream_fillbuf(stream, 1);

        ++stream->rwcnt_;
        return (int)(*stream->ptr_++);
}

static inline int jas_stream_getc_macro(jas_stream_t *stream)
{
        if (stream->flags_ & (JAS_STREAM_ERR | JAS_STREAM_EOF | JAS_STREAM_RWLIMIT))
                return EOF;

        if (stream->rwlimit_ >= 0 && stream->rwcnt_ >= stream->rwlimit_) {
                stream->flags_ |= JAS_STREAM_RWLIMIT;
                return EOF;
        }

        return jas_stream_getc2(stream);
}

/* Write a character to a stream. */
static inline int jas_stream_putc2(jas_stream_t *stream, jas_uchar c)
{
        stream->bufmode_ |= JAS_STREAM_WRBUF;

        if (--stream->cnt_ < 0)
                return jas_stream_flushbuf(stream, c);
        else {
                ++stream->rwcnt_;
                return (int)(*stream->ptr_++ = c);
        }
}

static inline int jas_stream_putc_macro(jas_stream_t *stream, jas_uchar c)
{
        if (stream->flags_ & (JAS_STREAM_ERR | JAS_STREAM_EOF | JAS_STREAM_RWLIMIT))
            return EOF;

        if (stream->rwlimit_ >= 0 && stream->rwcnt_ >= stream->rwlimit_) {
                stream->flags_ |= JAS_STREAM_RWLIMIT;
                return EOF;
        }

        return jas_stream_putc2(stream, c);
}

#ifdef __cplusplus
}
#endif

#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.

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_STRING_H
#define JAS_STRING_H

/*****************************************************************************\
* Includes.
\*****************************************************************************/

/* The configuration header file should be included first. */
#include <jasper/jas_config.h>

#include <jasper/jas_types.h>

#ifdef __cplusplus
extern "C" {
#endif

/*****************************************************************************\
* Functions.
\*****************************************************************************/

JAS_EXPORT
char *jas_strdup(const char *);

JAS_EXPORT
char *jas_strtok(char *str, const char *delim, char **saveptr);

JAS_EXPORT
int jas_stringtokenize(const char *string, const char *delim,
  char ***tokens_buf, size_t *max_tokens_buf, size_t *num_tokens_buf);

#ifdef __cplusplus
}
#endif

#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.

```
/*
 * Copyright (c) 1999-2000 Image Power, Inc. and the University of
 *   British Columbia.
 * Copyright (c) 2001-2003 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_THREAD_H
#define JAS_THREAD_H

/*****************************************************************************\
* Includes
\*****************************************************************************/

/* The configuration header file should be included first. */
#include <jasper/jas_config.h>

#include "jasper/jas_compiler.h"
#include "jasper/jas_types.h"

#if defined(JAS_THREADS)

#include <stdlib.h>
#include <assert.h>

#if defined(JAS_THREADS_C11)
#include <threads.h>
#include <stdatomic.h>
#elif defined(JAS_THREADS_PTHREAD)
#include <pthread.h>
#include <sched.h>
#elif defined(JAS_THREADS_WIN32)
#include <process.h>
#include <windows.h>
#include <processthreadsapi.h>
#endif

#endif

/*****************************************************************************\
\*****************************************************************************/

#ifdef __cplusplus
extern "C" {
#endif

#if defined(JAS_THREADS)
#if defined(JAS_FOR_INTERNAL_USE_ONLY) || defined(JAS_FOR_JASPER_APP_USE_ONLY)

/*****************************************************************************\
* Types
\*****************************************************************************/

#if defined(JAS_THREADS_C11)
#       define JAS_THREADS_IMPL "C11"
#       define JAS_USE_SPINLOCK
#elif defined(JAS_THREADS_PTHREAD)
#       define JAS_THREADS_IMPL "PTHREAD"
#       undef JAS_USE_SPINLOCK
#elif defined(JAS_THREADS_WIN32)
#       define JAS_THREADS_IMPL "WIN32"
#       define JAS_USE_SPINLOCK
#endif

/*************************************\
* Spinlock
\*************************************/

#if defined(JAS_THREADS_C11)
#       define JAS_USE_SPINLOCK
typedef struct {
        atomic_flag flag;
} jas_spinlock_t;
#elif defined(JAS_THREADS_PTHREAD)
/* There is no pthread_spinlock_t type on MacOS. */
#       undef JAS_USE_SPINLOCK
#elif defined(JAS_THREADS_WIN32)
#       define JAS_USE_SPINLOCK
```

```
typedef struct {
        LONG flag;
} jas_spinlock_t;
#endif

#if defined(JAS_THREADS_C11)
#define JAS_SPINLOCK_INITIALIZER {ATOMIC_FLAG_INIT}
#elif defined(JAS_THREADS_PTHREAD)
/* There is no pthread_spinlock_t type on MacOS. */
#elif defined(JAS_THREADS_WIN32)
#define JAS_SPINLOCK_INITIALIZER {0}
#endif

/*************************************\
* Basic Mutex
\*************************************/

#if defined(JAS_THREADS_C11)
typedef mtx_t jas_basicmutex_t;
#elif defined(JAS_THREADS_PTHREAD)
typedef pthread_mutex_t jas_basicmutex_t;
#elif defined(JAS_THREADS_WIN32)
typedef CRITICAL_SECTION jas_basicmutex_t;
#endif

#if defined(JAS_THREADS_C11)
#undef JAS_BASICMUTEX_INITIALIZER
#elif defined(JAS_THREADS_PTHREAD)
#define JAS_BASICMUTEX_INITIALIZER PTHREAD_MUTEX_INITIALIZER
#elif defined(JAS_THREADS_WIN32)
#define JAS_BASICMUTEX_INITIALIZER
#endif

/*************************************\
* Mutex (Allowing Static Initialization)
\*************************************/

#if defined(JAS_USE_SPINLOCK)
#       define jas_mutex_t jas_spinlock_t
#       define JAS_MUTEX_INITIALIZER JAS_SPINLOCK_INITIALIZER
#       define jas_mutex_init jas_spinlock_init
#       define jas_mutex_cleanup jas_spinlock_cleanup
#       define jas_mutex_lock jas_spinlock_lock
#       define jas_mutex_unlock jas_spinlock_unlock
#else
#       define jas_mutex_t jas_basicmutex_t
#       define JAS_MUTEX_INITIALIZER JAS_BASICMUTEX_INITIALIZER
#       define jas_mutex_init jas_basicmutex_init
#       define jas_mutex_cleanup jas_basicmutex_cleanup
#       define jas_mutex_lock jas_basicmutex_lock
#       define jas_mutex_unlock jas_basicmutex_unlock
#endif

/*************************************\
* Once Flag
\*************************************/

#if defined(JAS_THREADS_C11)
typedef once_flag jas_once_flag_t;
#elif defined(JAS_THREADS_PTHREAD)
typedef pthread_once_t jas_once_flag_t;
#elif defined(JAS_THREADS_WIN32)
typedef struct {
        volatile LONG status;
} jas_once_flag_t;
#endif

#if defined(JAS_THREADS_C11)
#       define JAS_ONCE_FLAG_INIT ONCE_FLAG_INIT
#elif defined(JAS_THREADS_PTHREAD)
#       define JAS_ONCE_FLAG_INIT PTHREAD_ONCE_INIT
#elif defined(JAS_THREADS_WIN32)
#       define JAS_ONCE_FLAG_INIT {0}
#endif

/*************************************\
* Threads
\*************************************/

#if defined(JAS_FOR_INTERNAL_USE_ONLY) || defined(JAS_FOR_JASPER_APP_USE_ONLY)

```

```
#if defined(JAS_THREADS_C11)
typedef thrd_t jas_thread_id_t;
#elif defined(JAS_THREADS_PTHREAD)
typedef pthread_t jas_thread_id_t;
#elif defined(JAS_THREADS_WIN32)
typedef HANDLE jas_thread_id_t;
#endif

#if defined(JAS_THREADS_C11)
typedef thrd_t jas_thread_t;
#elif defined(JAS_THREADS_PTHREAD)
typedef struct {
        pthread_t id;
        int (*func)(void *);
        void *arg;
        int result;
} jas_thread_t;
#elif defined(JAS_THREADS_WIN32)
typedef struct {
        jas_thread_id_t id;
        int (*func)(void *);
        void *arg;
} jas_thread_t;
#endif

static inline void jas_thread_yield(void);

#endif

/**************************************\
* Thread-Specific Storage (TSS)
\**************************************/

#if defined(JAS_THREADS_C11)
typedef tss_t jas_tss_t;
#elif defined(JAS_THREADS_PTHREAD)
typedef pthread_key_t jas_tss_t;
#elif defined(JAS_THREADS_WIN32)
typedef DWORD jas_tss_t;
#endif


/******************************************************************************\
* Spinlock
\******************************************************************************/

#if defined(JAS_USE_SPINLOCK)

static inline int jas_spinlock_init(jas_spinlock_t *mtx)
{
        assert(mtx);
#if defined(JAS_THREADS_C11)
        atomic_flag_clear(&mtx->flag);
        return 0;
#elif defined(JAS_THREADS_PTHREAD)
        JAS_UNUSED(mtx);
        abort();
        return -1;
#elif defined(JAS_THREADS_WIN32)
        InterlockedExchange(&mtx->flag, 0);
        return 0;
#endif
}

static inline int jas_spinlock_cleanup(jas_spinlock_t *mtx)
{
        assert(mtx);
#if defined(JAS_THREADS_C11)
        JAS_UNUSED(mtx);
        return 0;
#elif defined(JAS_THREADS_PTHREAD)
        JAS_UNUSED(mtx);
        abort();
        return -1;
#elif defined(JAS_THREADS_WIN32)
        JAS_UNUSED(mtx);
        return 0;
#endif
}

static inline int jas_spinlock_lock(jas_spinlock_t *mtx)
```

```
{
        assert(mtx);
#if defined(JAS_THREADS_C11)
        while (atomic_flag_test_and_set(&mtx->flag)) {}
        return 0;
#elif defined(JAS_THREADS_PTHREAD)
        JAS_UNUSED(mtx);
        abort();
        return -1;
#elif defined(JAS_THREADS_WIN32)
        while (InterlockedCompareExchange(&mtx->flag, 1, 0)) {}
        return 0;
#endif
}

static inline int jas_spinlock_unlock(jas_spinlock_t *mtx)
{
        assert(mtx);
#if defined(JAS_THREADS_C11)
        atomic_flag_clear(&mtx->flag);
        return 0;
#elif defined(JAS_THREADS_PTHREAD)
        JAS_UNUSED(mtx);
        abort();
        return -1;
#elif defined(JAS_THREADS_WIN32)
        InterlockedExchange(&mtx->flag, 0);
        return 0;
#endif
}

#endif

/*****************************************************************************\
* Basic Mutex
\*****************************************************************************/

/* For internal use only. */
static inline int jas_basicmutex_init(jas_basicmutex_t *mtx)
{
        assert(mtx);
#if defined(JAS_THREADS_C11)
        return mtx_init(mtx, mtx_plain) == thrd_success ? 0 : -1;
#elif defined(JAS_THREADS_PTHREAD)
        return pthread_mutex_init(mtx, 0);
#elif defined(JAS_THREADS_WIN32)
        InitializeCriticalSection(mtx);
        return 0;
#endif
}

/* For internal use only. */
static inline int jas_basicmutex_cleanup(jas_basicmutex_t *mtx)
{
        assert(mtx);
#if defined(JAS_THREADS_C11)
        mtx_destroy(mtx);
        return 0;
#elif defined(JAS_THREADS_PTHREAD)
        return pthread_mutex_destroy(mtx);
#elif defined(JAS_THREADS_WIN32)
        DeleteCriticalSection(mtx);
        return 0;
#endif
}

/* For internal use only. */
static inline int jas_basicmutex_lock(jas_basicmutex_t *mtx)
{
        assert(mtx);
#if defined(JAS_THREADS_C11)
        return mtx_lock(mtx);
#elif defined(JAS_THREADS_PTHREAD)
        return pthread_mutex_lock(mtx);
#elif defined(JAS_THREADS_WIN32)
        EnterCriticalSection(mtx);
        return 0;
#endif
}

/* For internal use only. */
```

```
static inline int jas_basicmutex_unlock(jas_basicmutex_t *mtx)
{
        assert(mtx);
#if defined(JAS_THREADS_C11)
        return mtx_unlock(mtx);
#elif defined(JAS_THREADS_PTHREAD)
        return pthread_mutex_unlock(mtx);
#elif defined(JAS_THREADS_WIN32)
        LeaveCriticalSection(mtx);
        return 0;
#endif
}

/*****************************************************************************\
* Thread-Specific Storage (TSS)
\*****************************************************************************/

static inline
int jas_tss_create(jas_tss_t *tss, void (*destructor)(void *))
{
        assert(tss);
#if defined(JAS_THREADS_C11)
        return tss_create(tss, destructor) == thrd_success ? 0 : -1;
#elif defined(JAS_THREADS_PTHREAD)
        return pthread_key_create(tss, destructor);
#elif defined(JAS_THREADS_WIN32)
        if (destructor) {
                return -1;
        }
        DWORD id;
        if ((id = TlsAlloc()) == TLS_OUT_OF_INDEXES) {
                return -2;
        }
        *tss = id;
        return 0;
#endif
}

static inline
void jas_tss_delete(jas_tss_t tss)
{
#if defined(JAS_THREADS_C11)
        tss_delete(tss);
#elif defined(JAS_THREADS_PTHREAD)
        pthread_key_delete(tss);
#elif defined(JAS_THREADS_WIN32)
        TlsFree(tss);
#endif
}

static inline
void *jas_tss_get(jas_tss_t tss)
{
#if defined(JAS_THREADS_C11)
        return tss_get(tss);
#elif defined(JAS_THREADS_PTHREAD)
        return pthread_getspecific(tss);
#elif defined(JAS_THREADS_WIN32)
        return TlsGetValue(tss);
#endif
}

static inline
int jas_tss_set(jas_tss_t tss, void *value)
{
#if defined(JAS_THREADS_C11)
        return tss_set(tss, value) == thrd_success ? 0 : -1;
#elif defined(JAS_THREADS_PTHREAD)
        return pthread_setspecific(tss, value);
#elif defined(JAS_THREADS_WIN32)
        return TlsSetValue(tss, value) ? 0 : -1;
#endif
}

/*****************************************************************************\
* Once Flag
\*****************************************************************************/

static inline int jas_call_once(jas_once_flag_t *flag, void (*func)(void))
{
        assert(flag);
```

```
        assert(func);
#if defined(JAS_THREADS_C11)
        call_once(flag, func);
        return 0;
#elif defined(JAS_THREADS_PTHREAD)
        return pthread_once(flag, func);
#elif defined(JAS_THREADS_WIN32)
        if (InterlockedCompareExchange(&flag->status, 1, 0) == 0) {
                (func)();
                InterlockedExchange(&flag->status, 2);
        } else {
                while (flag->status == 1) {
                        /* Perform a busy wait.  This is ugly. */
                        /* Yield processor. */
                        SwitchToThread();
                }
        }
        return 0;
#endif
}

/*****************************************************************************\
* Threads
\*****************************************************************************/

#if defined(JAS_FOR_INTERNAL_USE_ONLY) || defined(JAS_FOR_JASPER_APP_USE_ONLY)

#if defined(JAS_THREADS_PTHREAD)
static void *thread_func_wrapper(void *thread_ptr)
{
        jas_thread_t *thread = JAS_CAST(jas_thread_t *, thread_ptr);
        int result = (thread->func)(thread->arg);
        thread->result = result;
        return thread;
}
#elif defined(JAS_THREADS_WIN32)
static unsigned __stdcall thread_func_wrapper(void *thread_ptr)
{
        jas_thread_t *thread = JAS_CAST(jas_thread_t *, thread_ptr);
        int result = (thread->func)(thread->arg);
        return JAS_CAST(unsigned, result);
}
#endif

static inline
int jas_thread_compare(jas_thread_id_t x, jas_thread_id_t y)
{
#if defined(JAS_THREADS_C11)
        return thrd_equal(x, y);
#elif defined(JAS_THREADS_PTHREAD)
        return pthread_equal(x, y);
#elif defined(JAS_THREADS_WIN32)
        return GetThreadId(x) == GetThreadId(y);
#endif
}

static inline
int jas_thread_create(jas_thread_t *thread, int (*func)(void *), void *arg)
{
        assert(thread);
        assert(func);
#if defined(JAS_THREADS_C11)
        return thrd_create(thread, func, arg) == thrd_success ? 0 : -1;
#elif defined(JAS_THREADS_PTHREAD)
        thread->func = func;
        thread->arg = arg;
        thread->result = 0;
        return pthread_create(&thread->id, 0, thread_func_wrapper, thread);
#elif defined(JAS_THREADS_WIN32)
        uintptr_t handle;
        thread->func = func;
        thread->arg = arg;
        if (!(handle = _beginthreadex(0, 0, thread_func_wrapper, thread, 0, 0))) {
                return -1;
        }
        thread->id = JAS_CAST(jas_thread_id_t, handle);
        return 0;
#endif
}

static inline
```

```
int jas_thread_join(jas_thread_t *thread, int *result)
{
        assert(thread);
#if defined(JAS_THREADS_C11)
        return thrd_join(*thread, result) == thrd_success ? 0 : -1;
#elif defined(JAS_THREADS_PTHREAD)
        void *result_buf;
        int ret = pthread_join(thread->id, &result_buf);
        if (!ret) {
                jas_thread_t *other_thread = JAS_CAST(jas_thread_t *, result_buf);
                if (result) {
                        /* A null pointer is probably a bug. */
                        assert(other_thread);
                        *result = other_thread ? other_thread->result : 0;
                }
        }
        return ret;
#elif defined(JAS_THREADS_WIN32)
        DWORD w;
        DWORD code;
        if ((w = WaitForSingleObject(thread->id, INFINITE)) != WAIT_OBJECT_0) {
                return -1;
        }
        if (result) {
                if (!GetExitCodeThread(thread->id, &code)) {
                        CloseHandle(thread->id);
                        return -1;
                }
                *result = JAS_CAST(int, code);
        }
        CloseHandle(thread->id);
        return 0;
#endif
}

static inline void jas_thread_yield(void)
{
#if defined(JAS_THREADS_C11)
        thrd_yield();
#elif defined(JAS_THREADS_PTHREAD)
        sched_yield();
#elif defined(JAS_THREADS_WIN32)
        SwitchToThread();
#endif
}

#if 0
/* This functionality is not available for all threading support libraries. */
static inline
void jas_thread_exit(int result)
{
#if defined(JAS_THREADS_C11)
        thrd_exit(result);
#elif defined(JAS_THREADS_PTHREAD)
        /* This does not have a trivial implementation, as far as I can see. */
        /* There is no jas_thread_find function. */
        jas_thread_t *thread = jas_thread_find(pthread_self());
        thread->result = result;
        pthread_exit(JAS_CAST(void *, thread));
#endif
}
#endif

#if 0
static inline
jas_thread_id_t jas_thread_current(void)
{
#if defined(JAS_THREADS_C11)
        return thrd_current();
#elif defined(JAS_THREADS_PTHREAD)
        return pthread_self();
#elif defined(JAS_THREADS_WIN32)
        /* FIXME - NOT YET IMPLEMENTED. */
        abort();
#endif
}
#endif

#endif

/*****************************************************************************\
```

```
*
\*****************************************************************************/

#endif
#else

/*****************************************************************************\
* No Threading Support.
\*****************************************************************************/

#endif

#ifdef __cplusplus
}
#endif

#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.

```
/*
 * Copyright (c) 2004 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_TMR_H
#define JAS_TMR_H

/* The configuration header file should be included first. */
#include <jasper/jas_config.h>

#if defined(JAS_HAVE_SYS_TIME_H)
#include <sys/time.h>
#else
#include <time.h>
#endif

#ifdef __cplusplus
extern "C" {
#endif

#if defined(JAS_HAVE_GETTIMEOFDAY)

typedef struct {
        struct timeval start;
        struct timeval stop;
} jas_tmr_t;

#elif defined(JAS_HAVE_GETRUSAGE)

typedef struct {
        struct rusage start;
        struct rusage stop;
} jas_tmr_t;

#else

```

```
typedef int jas_tmr_t;

#endif

JAS_EXPORT
void jas_tmr_start(jas_tmr_t *tmr);

JAS_EXPORT
void jas_tmr_stop(jas_tmr_t *tmr);

JAS_EXPORT
double jas_tmr_get(jas_tmr_t *tmr);

#ifdef __cplusplus
}
#endif

#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_nonull (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.

```
/*
 * Copyright (c) 2001-2002 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_TVP_H
#define JAS_TVP_H

/*****************************************************************************\
* Includes.
```

```
\*****************************************************************************/

/* The configuration header file should be included first. */
#include <jasper/jas_config.h>

#ifdef __cplusplus
extern "C" {
#endif

/*****************************************************************************\
* Types.
\*****************************************************************************/

typedef struct {

        int id;
        /* The ID for the tag. */

        const char *name;
         /* The name of the tag. */

} jas_taginfo_t;

typedef struct {

         char *buf;
         /* The parsing buffer. */

         char *tag;
         /* The current tag name. */

         const char *val;
         /* The current value. */

         char *pos;
         /* The current position in the parsing buffer. */

} jas_tvparser_t;

/*****************************************************************************\
* Tag information functions.
\*****************************************************************************/

JAS_ATTRIBUTE_PURE
JAS_EXPORT
const jas_taginfo_t *jas_taginfos_lookup(const jas_taginfo_t *taginfos,
  const char *name);

JAS_ATTRIBUTE_PURE
JAS_EXPORT
const jas_taginfo_t *jas_taginfo_nonull(const jas_taginfo_t *taginfo);

/*****************************************************************************\
* Tag-value parser functions.
\*****************************************************************************/

JAS_EXPORT
jas_tvparser_t *jas_tvparser_create(const char *s);

JAS_EXPORT
void jas_tvparser_destroy(jas_tvparser_t *tvparser);

JAS_EXPORT
int jas_tvparser_next(jas_tvparser_t *tvparser);

JAS_ATTRIBUTE_PURE
JAS_EXPORT
const char *jas_tvparser_gettag(const jas_tvparser_t *tvparser);

JAS_ATTRIBUTE_PURE
JAS_EXPORT
const char *jas_tvparser_getval(const jas_tvparser_t *tvparser);

#ifdef __cplusplus
}
#endif

#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.

```
/*
 * Copyright (c) 1999-2000 Image Power, Inc. and the University of
 *   British Columbia.
 * Copyright (c) 2001-2003 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_TYPES_H
#define JAS_TYPES_H

/* The configuration header file should be included first. */
#include <jasper/jas_config.h>

/* Note: The immediately following header files should eventually be removed. */
#include <stddef.h> /* IWYU pragma: export */
#include <stdint.h> /* IWYU pragma: export */

#include <limits.h> /* IWYU pragma: export */

#if defined(JAS_HAVE_SYS_TYPES_H)
#include <sys/types.h> /* IWYU pragma: export */
#endif

#ifdef __cplusplus
extern "C" {
#endif

#define jas_uchar unsigned char
#define jas_uint unsigned int
#define jas_ulong unsigned long
#define jas_longlong long long
#define jas_ulonglong unsigned long long

#if !defined(JAS_NO_SET_SSIZE_T)
#       if !defined(SSIZE_MAX)
#               if (JAS_SIZEOF_INT == JAS_SIZEOF_SIZE_T)
#                       define ssize_t int
#                       define SSIZE_MAX INT_MAX
#               elif (JAS_SIZEOF_LONG == JAS_SIZEOF_SIZE_T)
#                       define ssize_t long
#                       define SSIZE_MAX LONG_MAX
#               else
#                       define ssize_t jas_longlong
#                       define SSIZE_MAX LLONG_MAX
#               endif
#       endif
#endif

#if 0
#if defined(JAS_HAVE_SSIZE_T)
#define jas_ssize_t ssize_t
#define JAS_SSIZE_MAX SSIZE_MAX
#else
#define jas_ssize_t jas_longlong
#define JAS_SSIZE_MAX LLONG_MAX
#endif
#endif

#if defined(_MSC_VER) && (_MSC_VER < 1800)
#define bool  int
#define false 0
#define true  1

#define PRIxFAST32 "x"
#define PRIxFAST16 PRIxFAST32
#define PRIuFAST32 "u"
#define PRIuFAST16 PRIuFAST32
#define PRIiFAST32 "i"
#ifdef _WIN64
    #define PRIdPTR "lld"
```

```
#else
    #define PRIdPTR "d"
#endif

#ifndef _HUGE_ENUF
    #define _HUGE_ENUF 1e+300
#endif

#define INFINITY ((float)(_HUGE_ENUF * _HUGE_ENUF))

#define strtoull _strtoui64

#else
#include <stdbool.h> /* IWYU pragma: export */
#include <inttypes.h> /* IWYU pragma: export */
#endif

/* The below macro is intended to be used for type casts.  By using this
  macro, type casts can be easily located in the source code with
  tools like "grep". */
#define JAS_CAST(t, e) \
        ((t) (e))

/* The number of bits in the integeral type uint_fast32_t. */
/* NOTE: This could underestimate the size on some exotic architectures. */
#define JAS_UINTFAST32_NUMBITS (8 * sizeof(uint_fast32_t))

#if 0
#if defined(JAS_HAVE_MAX_ALIGN_T)
#define jas_max_align_t max_align_t
#else
#define jas_max_align_t long double
#endif
#endif

/*
Assume that a compiler claiming to be compliant with C11 or a later version
of the C standard provides a suitable definition of max_align_t.
The JAS_NO_SET_MAX_ALIGN_T preprocessor symbol can be used to override
this behavior.
*/
#if defined(JAS_NO_SET_MAX_ALIGN_T)
        /*
        The user of this header is assuming responsibility for providing a
        suitable definition for max_align_t.
        */
#elif defined(_MSC_VER)
        /*
        Define max_align_t as a preprocessor symbol since using typedef will
        cause problems.
        */
#       define max_align_t long double
#elif !(defined(__STDC_VERSION__) && (__STDC_VERSION__ - 0 >= 201112L))
#       define max_align_t long double
#endif

#if 0
#if defined(JAS_HAVE_UINTMAX_T)
#define jas_uintmax_t uintmax_t
#else
#define jas_uintmax_t uint_fast64_t
#endif
#endif

#if 0
#if defined(JAS_HAVE_INTMAX_T)
#define jas_intmax_t intmax_t
#else
#define jas_intmax_t int_fast64_t
#endif
#endif

#ifdef __cplusplus
}
#endif

#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.

```
/*
 * Copyright (c) 1999-2000 Image Power, Inc. and the University of
 *   British Columbia.
 * Copyright (c) 2001-2003 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_VERSION_H
#define JAS_VERSION_H

/* The configuration header file should be included first. */
#include <jasper/jas_config.h>

#ifdef __cplusplus
extern "C" {
#endif

/*****************************************************************************\
* Constants and types.
\*****************************************************************************/

#define JAS_COPYRIGHT \
        "Copyright (c) 2001-2022 Michael David Adams.\n" \
        "Copyright (c) 1999-2000 Image Power, Inc. and the University of\n" \
        "  British Columbia.\n" \
        "All rights reserved.\n"

#define JAS_NOTES \
        "For more information about this software, please visit the following\n" \
        "web sites/pages:\n" \
        "    https://jasper-software.github.io/jasper\n" \
        "    https://jasper-software.github.io/jasper-manual\n" \
        "    https://www.ece.uvic.ca/~mdadams/jasper\n" \
        "Please submit bug reports using the bug-tracker at:\n" \
        "    https://github.com/jasper-software/jasper/issues\n"

/*****************************************************************************\
* Functions.
\*****************************************************************************/

JAS_ATTRIBUTE_CONST
JAS_EXPORT
const char *jas_getversion(void);
/* Get the version information for the JasPer library. */
/* Note:  Since libjasper can be built as a shared library, the version
  returned by this function may not necessarily correspond to JAS_VERSION. */

#ifdef __cplusplus
}
#endif

#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.

```
/*
 * Copyright (c) 2001-2003 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JAS_JASPER_H
#define JAS_JASPER_H

// IWYU pragma: begin_exports

/* The configuration header file should be included first. */
#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_init.h>
#include <jasper/jas_log.h>

#ifdef __cplusplus
extern "C" {
#endif

#ifdef __cplusplus
}
#endif

// IWYU pragma: end_exports

#endif
```

16.45 jp2_cod.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * JP2 Library
 *
 * $Id$
 */

#ifndef JP2_COD_H
#define JP2_COD_H

/*****************************************************************************\
* Includes.
\*****************************************************************************/

#include "jasper/jas_types.h"
#include "jasper/jas_stream.h"
#include "jasper/jas_image.h"

#include <stdio.h>

/*****************************************************************************\
* Macros.
\*****************************************************************************/

#define JP2_SPTOBPC(s, p) \
        ((((p) - 1) & 0x7f) | (((s) & 1) « 7))

/*****************************************************************************\
* Box class.
\*****************************************************************************/

#define JP2_BOX_HDRLEN(ext) ((ext) ? 16 : 8)

/* Box types. */
#define JP2_BOX_JP              0x6a502020      /* Signature */
#define JP2_BOX_FTYP    0x66747970      /* File Type */
#define JP2_BOX_JP2H    0x6a703268      /* JP2 Header */
#define JP2_BOX_IHDR    0x69686472      /* Image Header */
```

```
#define JP2_BOX_BPCC    0x62706363      /* Bits Per Component */
#define JP2_BOX_COLR    0x636f6c72      /* Color Specification */
#define JP2_BOX_PCLR    0x70636c72      /* Palette */
#define JP2_BOX_CMAP    0x636d6170      /* Component Mapping */
#define JP2_BOX_CDEF    0x63646566      /* Channel Definition */
#define JP2_BOX_RES             0x72657320      /* Resolution */
#define JP2_BOX_RESC    0x72657363      /* Capture Resolution */
#define JP2_BOX_RESD    0x72657364      /* Default Display Resolution */
#define JP2_BOX_JP2C    0x6a703263      /* Contiguous Code Stream */
#define JP2_BOX_JP2I    0x6a703269      /* Intellectual Property */
#define JP2_BOX_XML             0x786d6c20      /* XML */
#define JP2_BOX_UUID    0x75756964      /* UUID */
#define JP2_BOX_UINF    0x75696e66      /* UUID Info */
#define JP2_BOX_ULST    0x75637374      /* UUID List */
#define JP2_BOX_URL             0x75726c20      /* URL */

#define JP2_BOX_SUPER   0x01
#define JP2_BOX_NODATA  0x02

/* JP box data. */

#define JP2_JP_MAGIC    0x0d0a870a
#define JP2_JP_LEN              12

typedef struct {
        uint_fast32_t magic;
} jp2_jp_t;

/* FTYP box data. */

#define JP2_FTYP_MAXCOMPATCODES 32
#define JP2_FTYP_MAJVER         0x6a703220
#define JP2_FTYP_MINVER         0
#define JP2_FTYP_COMPATCODE             JP2_FTYP_MAJVER

typedef struct {
        uint_fast32_t majver;
        uint_fast32_t minver;
        uint_fast32_t numcompatcodes;
        uint_fast32_t compatcodes[JP2_FTYP_MAXCOMPATCODES];
} jp2_ftyp_t;

/* IHDR box data. */

#define JP2_IHDR_COMPTYPE       7
#define JP2_IHDR_BPCNULL        255

typedef struct {
        uint_fast32_t width;
        uint_fast32_t height;
        uint_fast16_t numcmpts;
        uint_fast8_t bpc;
        uint_fast8_t comptype;
        uint_fast8_t csunk;
        uint_fast8_t ipr;
} jp2_ihdr_t;

/* BPCC box data. */

typedef struct {
        uint_fast16_t numcmpts;
        uint_fast8_t *bpcs;
} jp2_bpcc_t;

/* COLR box data. */

#define JP2_COLR_ENUM   1
#define JP2_COLR_ICC    2
#define JP2_COLR_PRI    0

#define JP2_COLR_SRGB   16
#define JP2_COLR_SGRAY  17
#define JP2_COLR_SYCC   18

typedef struct {
        uint_fast8_t method;
        uint_fast8_t pri;
        uint_fast8_t approx;
        uint_fast32_t csid;
        uint_fast8_t *iccp;
        size_t iccplen;
```

```
        /* XXX - Someday we ought to add ICC profile data here. */
} jp2_colr_t;

/* PCLR box data. */

typedef struct {
        uint_fast16_t numlutents;
        uint_fast8_t numchans;
        int_fast32_t *lutdata;
        uint_fast8_t *bpc;
} jp2_pclr_t;

/* CDEF box per-channel data. */

#define JP2_CDEF_RGB_R  1
#define JP2_CDEF_RGB_G  2
#define JP2_CDEF_RGB_B  3

#define JP2_CDEF_YCBCR_Y        1
#define JP2_CDEF_YCBCR_CB       2
#define JP2_CDEF_YCBCR_CR       3

#define JP2_CDEF_GRAY_Y 1

#define JP2_CDEF_TYPE_COLOR     0
#define JP2_CDEF_TYPE_OPACITY   1
#define JP2_CDEF_TYPE_UNSPEC    65535
#define JP2_CDEF_ASOC_ALL       0
#define JP2_CDEF_ASOC_NONE      65535

typedef struct {
        uint_fast16_t channo;
        uint_fast16_t type;
        uint_fast16_t assoc;
} jp2_cdefchan_t;

/* CDEF box data. */

typedef struct {
        uint_fast16_t numchans;
        jp2_cdefchan_t *ents;
} jp2_cdef_t;

typedef struct {
        uint_fast16_t cmptno;
        uint_fast8_t map;
        uint_fast8_t pcol;
} jp2_cmapent_t;

typedef struct {
        uint_fast16_t numchans;
        jp2_cmapent_t *ents;
} jp2_cmap_t;

#define JP2_CMAP_DIRECT         0
#define JP2_CMAP_PALETTE        1

/* Generic box. */

struct jp2_boxops_s;
typedef struct {

        const struct jp2_boxops_s *ops;
        const struct jp2_boxinfo_s *info;

        uint_fast32_t type;

        /* The length of the box including the (variable-length) header. */
        uint_fast32_t len;

        /* The length of the box data. */
        uint_fast32_t datalen;

        union {
                jp2_jp_t jp;
                jp2_ftyp_t ftyp;
                jp2_ihdr_t ihdr;
                jp2_bpcc_t bpcc;
                jp2_colr_t colr;
                jp2_pclr_t pclr;
                jp2_cdef_t cdef;
```

```
                jp2_cmap_t cmap;
        } data;

} jp2_box_t;

typedef struct jp2_boxops_s {
        void (*init)(jp2_box_t *box);
        void (*destroy)(jp2_box_t *box);
        int (*getdata)(jp2_box_t *box, jas_stream_t *in);
        int (*putdata)(const jp2_box_t *box, jas_stream_t *out);
        void (*dumpdata)(const jp2_box_t *box);
} jp2_boxops_t;

/*****************************************************************************\
*
\*****************************************************************************/

typedef struct jp2_boxinfo_s {
        int type;
        int flags;
        const char *name;
        jp2_boxops_t ops;
} jp2_boxinfo_t;

/*****************************************************************************\
* Box class.
\*****************************************************************************/

jp2_box_t *jp2_box_create(int type);
void jp2_box_destroy(jp2_box_t *box);
jp2_box_t *jp2_box_get(jas_stream_t *in);
int jp2_box_put(jp2_box_t *box, jas_stream_t *out);

JAS_ATTRIBUTE_CONST
static inline uint_least8_t JP2_DTYPETOBPC(uint_least8_t dtype)
{
        return (JAS_IMAGE_CDT_GETSGND(dtype) « 7) | (JAS_IMAGE_CDT_GETPREC(dtype) - 1);
}

JAS_ATTRIBUTE_CONST
static inline uint_least8_t JP2_BPCTODTYPE(uint_least8_t bpc)
{
        return JAS_IMAGE_CDT_SETSGND(bpc » 7) | JAS_IMAGE_CDT_SETPREC((bpc & 0x7f) + 1);
}

#define ICC_CS_RGB      0x52474220
#define ICC_CS_YCBCR    0x59436272
#define ICC_CS_GRAY     0x47524159

const jp2_cdefchan_t *jp2_cdef_lookup(jp2_cdef_t *cdef, int channo);

#endif
```

16.46 jp2_dec.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JP2_DEC_H
#define JP2_DEC_H

#include "jasper/jas_image.h"
#include "jp2_cod.h"

typedef struct {

        jp2_box_t *pclr;
        jp2_box_t *cdef;
        jp2_box_t *ihdr;
        jp2_box_t *bpcc;
        jp2_box_t *cmap;
        jp2_box_t *colr;
        jas_image_t *image;
        uint_fast16_t numchans;
        uint_fast16_t *chantocmptlut;

} jp2_dec_t;

#endif
```

16.47 jpc_bs.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * Bit Stream Class
 *
 * $Id$
 */

#ifndef JPC_BS_H
#define JPC_BS_H

/******************************************************************************\
* Includes.
\******************************************************************************/

#include "jasper/jas_types.h"
#include "jasper/jas_stream.h"

#include <assert.h>
#include <stdio.h>

/******************************************************************************\
* Constants.
\******************************************************************************/

/*
 * Bit stream open mode flags.
 */

/* Bit stream open for reading. */
#define JPC_BITSTREAM_READ      0x01
/* Bit stream open for writing. */
```

```
#define JPC_BITSTREAM_WRITE     0x02

/*
 * Bit stream flags.
 */

/* Do not close underlying character stream. */
#define JPC_BITSTREAM_NOCLOSE   0x01
/* End of file has been reached while reading. */
#define JPC_BITSTREAM_EOF       0x02
/* An I/O error has occured. */
#define JPC_BITSTREAM_ERR       0x04

/******************************************************************************\
* Types.
\******************************************************************************/

/* Bit stream class. */

typedef struct {

        /* Some miscellaneous flags. */
        int flags_;

        /* The input/output buffer. */
        uint_fast16_t buf_;

        /* The number of bits remaining in the byte being read/written. */
        int cnt_;

        /* The underlying stream associated with this bit stream. */
        jas_stream_t *stream_;

        /* The mode in which this bit stream was opened. */
        int openmode_;

} jpc_bitstream_t;

/******************************************************************************\
* Functions/macros for opening and closing bit streams..
\******************************************************************************/

/* Open a stream as a bit stream. */
jpc_bitstream_t *jpc_bitstream_sopen(jas_stream_t *stream, const char *mode);

/* Close a bit stream. */
int jpc_bitstream_close(jpc_bitstream_t *bitstream);

/******************************************************************************\
* Functions/macros for reading from and writing to bit streams..
\******************************************************************************/

/* Read a bit from a bit stream. */
#ifndef NDEBUG
#define jpc_bitstream_getbit(bitstream) \
        jpc_bitstream_getbit_func(bitstream)
#else
#define jpc_bitstream_getbit(bitstream) \
        jpc_bitstream_getbit_macro(bitstream)
#endif

/* Write a bit to a bit stream. */
#ifndef NDEBUG
#define jpc_bitstream_putbit(bitstream, v) \
        jpc_bitstream_putbit_func(bitstream, v)
#else
#define jpc_bitstream_putbit(bitstream, v) \
        jpc_bitstream_putbit_macro(bitstream, v)
#endif

/* Read one or more bits from a bit stream. */
long jpc_bitstream_getbits(jpc_bitstream_t *bitstream, int n);

/* Write one or more bits to a bit stream. */
int jpc_bitstream_putbits(jpc_bitstream_t *bitstream, int n, long v);

/******************************************************************************\
* Functions/macros for flushing and aligning bit streams.
\******************************************************************************/

/* Align the current position within the bit stream to the next byte
```

```
  boundary. */
int jpc_bitstream_align(jpc_bitstream_t *bitstream);

/* Align the current position in the bit stream with the next byte boundary,
  ensuring that certain bits consumed in the process match a particular
  pattern. */
int jpc_bitstream_inalign(jpc_bitstream_t *bitstream, int fillmask,
  int filldata);

/* Align the current position in the bit stream with the next byte boundary,
  writing bits from the specified pattern (if necessary) in the process. */
int jpc_bitstream_outalign(jpc_bitstream_t *bitstream, int filldata);

/* Check if a bit stream needs alignment. */
JAS_ATTRIBUTE_PURE
int jpc_bitstream_needalign(const jpc_bitstream_t *bitstream);

/* How many additional bytes would be output if the bit stream was aligned? */
JAS_ATTRIBUTE_PURE
int jpc_bitstream_pending(const jpc_bitstream_t *bitstream);

/*****************************************************************************\
* Functions/macros for querying state information for bit streams.
\*****************************************************************************/

/* Has EOF been encountered on a bit stream? */
#define jpc_bitstream_eof(bitstream) \
        ((bitstream)->flags_ & JPC_BITSTREAM_EOF)

/*****************************************************************************\
* Internals.
\*****************************************************************************/

/* DO NOT DIRECTLY INVOKE ANY OF THE MACROS OR FUNCTIONS BELOW.  THEY ARE
  FOR INTERNAL USE ONLY. */

int jpc_bitstream_getbit_func(jpc_bitstream_t *bitstream);

int jpc_bitstream_putbit_func(jpc_bitstream_t *bitstream, int v);

int jpc_bitstream_fillbuf(jpc_bitstream_t *bitstream);

#define jpc_bitstream_getbit_macro(bitstream) \
        (assert((bitstream)->openmode_ & JPC_BITSTREAM_READ), \
          (--(bitstream)->cnt_ >= 0) ? \
          ((int)(((bitstream)->buf_ » (bitstream)->cnt_) & 1)) : \
          jpc_bitstream_fillbuf(bitstream))

#define jpc_bitstream_putbit_macro(bitstream, bit) \
        (assert((bitstream)->openmode_ & JPC_BITSTREAM_WRITE), \
          (--(bitstream)->cnt_ < 0) ? \
          ((bitstream)->buf_ = ((bitstream)->buf_ « 8) & 0xffff, \
          (bitstream)->cnt_ = ((bitstream)->buf_ == 0xff00) ? 6 : 7, \
          (bitstream)->buf_ |= ((bit) & 1) « (bitstream)->cnt_, \
          (jas_stream_putc((bitstream)->stream_, (bitstream)->buf_ » 8) == EOF) \
          ? (EOF) : ((bit) & 1)) : \
          ((bitstream)->buf_ |= ((bit) & 1) « (bitstream)->cnt_, \
          (bit) & 1))

#endif
```

16.48 jpc_cod.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * $Id$
 */

#ifndef JPC_COD_H
#define JPC_COD_H

#include "jpc_t1cod.h"

/******************************************************************************\
* Constants.
\******************************************************************************/

/* The nominal word size used by this implementation. */
#define JPC_PREC        32

void jpc_init(void);

#endif
```

16.49 jpc_cs.h

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

```

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * JPEG-2000 Code Stream Library
 *
 * $Id$
 */

#ifndef JPC_CS_H
#define JPC_CS_H

/******************************************************************************\
* Includes.
\******************************************************************************/

#include "jasper/jas_stream.h"

#include <assert.h>
#include <stdio.h>

/******************************************************************************\
* Constants and Types.
\******************************************************************************/

/* The maximum number of resolution levels. */
#define JPC_MAXRLVLS    33
```

```
/* The maximum number of bands. */
#define JPC_MAXBANDS    (3 * JPC_MAXRLVLS + 1)

/* The maximum number of layers. */
#define JPC_MAXLYRS     16384

/***************************************\
* Code stream.
\***************************************/

/*
 * Code stream states.
 */

/* Initial. */
#define JPC_CS_INIT     0
/* Main header. */
#define JPC_CS_MHDR     1
/* Tile-part header. */
#define JPC_CS_THDR     2
/* Main trailer. */
#define JPC_CS_MTLR     3
/* Tile-part data. */
#define JPC_CS_TDATA    4

/*
 * Unfortunately, the code stream syntax was not designed in such a way that
 * any given marker segment can be correctly decoded without additional state
 * derived from previously decoded marker segments.
 * For example, a RGN/COC/QCC marker segment cannot be decoded unless the
 * number of components is known.
 */

/*
 * Code stream state information.
 */

typedef struct {

        /* The number of components. */
        uint_fast16_t numcomps;

} jpc_cstate_t;

/***************************************\
* SOT marker segment parameters.
\***************************************/

typedef struct {

        /* The tile number. */
        uint_fast16_t tileno;

        /* The combined length of the marker segment and its auxilary data
          (i.e., packet data). */
        uint_fast32_t len;

        /* The tile-part instance. */
        uint_fast8_t partno;

        /* The number of tile-parts. */
        uint_fast8_t numparts;

} jpc_sot_t;

/***************************************\
* SIZ marker segment parameters.
\***************************************/

/* Per component information. */

typedef struct {

        /* The precision of the samples. */
        uint_fast8_t prec;

        /* The signedness of the samples. */
        uint_fast8_t sgnd;

        /* The horizontal separation of samples with respect to the reference
```

```
          grid. */
        uint_fast8_t hsamp;

        /* The vertical separation of samples with respect to the reference
          grid. */
        uint_fast8_t vsamp;

} jpc_sizcomp_t;

/* SIZ marker segment parameters. */

typedef struct {

        /* The code stream capabilities. */
        uint_fast16_t caps;

        /* The width of the image in units of the reference grid. */
        uint_fast32_t width;

        /* The height of the image in units of the reference grid. */
        uint_fast32_t height;

        /* The horizontal offset from the origin of the reference grid to the
          left side of the image area. */
        uint_fast32_t xoff;

        /* The vertical offset from the origin of the reference grid to the
          top side of the image area. */
        uint_fast32_t yoff;

        /* The nominal width of a tile in units of the reference grid. */
        uint_fast32_t tilewidth;

        /* The nominal height of a tile in units of the reference grid. */
        uint_fast32_t tileheight;

        /* The horizontal offset from the origin of the reference grid to the
          left side of the first tile. */
        uint_fast32_t tilexoff;

        /* The vertical offset from the origin of the reference grid to the
          top side of the first tile. */
        uint_fast32_t tileyoff;

        /* The number of components. */
        uint_fast16_t numcomps;

        /* The per-component information. */
        jpc_sizcomp_t *comps;

} jpc_siz_t;

/**************************************\
* COD marker segment parameters.
\**************************************/

/*
 * Coding style constants.
 */

/* Precincts may be used. */
#define JPC_COX_PRT     0x01
/* SOP marker segments may be used. */
#define JPC_COD_SOP     0x02
/* EPH marker segments may be used. */
#define JPC_COD_EPH     0x04

/*
 * Progression order constants.
 */

/* Layer-resolution-component-precinct progressive
  (i.e., progressive by fidelity). */
#define JPC_COD_LRCPPRG 0
/* Resolution-layer-component-precinct progressive
  (i.e., progressive by resolution). */
#define JPC_COD_RLCPPRG 1
/* Resolution-precinct-component-layer progressive. */
#define JPC_COD_RPCLPRG 2
/* Precinct-component-resolution-layer progressive. */
#define JPC_COD_PCRLPRG 3
```

```
/* Component-position-resolution-layer progressive. */
#define JPC_COD_CPRLPRG 4

/*
 * Code block style constants.
 */

#define JPC_COX_LAZY    0x01 /* Selective arithmetic coding bypass. */
#define JPC_COX_RESET   0x02 /* Reset context probabilities. */
#define JPC_COX_TERMALL 0x04 /* Terminate all coding passes. */
#define JPC_COX_VSC             0x08 /* Vertical stripe causal context formation. */
#define JPC_COX_PTERM   0x10 /* Predictable termination. */
#define JPC_COX_SEGSYM  0x20 /* Use segmentation symbols. */

/* Transform constants. */
#define JPC_COX_INS     0x00 /* Irreversible 9/7. */
#define JPC_COX_RFT     0x01 /* Reversible 5/3. */

/* Multicomponent transform constants. */
#define JPC_COD_NOMCT   0x00 /* No multicomponent transform. */
#define JPC_COD_MCT             0x01 /* Multicomponent transform. */

/* Get the code block size value from the code block size exponent. */
JAS_ATTRIBUTE_CONST
static inline unsigned JPC_COX_CBLKSIZEEXPN(unsigned x)
{
        return x - 2;
}

/* Get the code block size exponent from the code block size value. */
JAS_ATTRIBUTE_CONST
static inline unsigned JPC_COX_GETCBLKSIZEEXPN(unsigned x)
{
        return x + 2;
}

/* Per resolution-level information. */

typedef struct {

        /* The packet partition width. */
        uint_fast8_t parwidthval;

        /* The packet partition height. */
        uint_fast8_t parheightval;

} jpc_coxrlvl_t;

/* Per component information. */

typedef struct {

        /* The coding style. */
        uint_fast8_t csty;

        /* The number of decomposition levels. */
        uint_fast8_t numdlvls;

        /* The nominal code block width specifier. */
        uint_fast8_t cblkwidthval;

        /* The nominal code block height specifier. */
        uint_fast8_t cblkheightval;

        /* The style of coding passes. */
        uint_fast8_t cblksty;

        /* The QMFB employed. */
        uint_fast8_t qmfbid;

        /* The number of resolution levels. */
        int numrlvls;

        /* The per-resolution-level information. */
        jpc_coxrlvl_t rlvls[JPC_MAXRLVLS];

} jpc_coxcp_t;

/* COD marker segment parameters. */

typedef struct {
```

```

        /* The general coding style. */
        uint_fast8_t csty;

        /* The progression order. */
        uint_fast8_t prg;

        /* The number of layers. */
        uint_fast16_t numlyrs;

        /* The multicomponent transform. */
        uint_fast8_t mctrans;

        /* Component-related parameters. */
        jpc_coxcp_t compparms;

} jpc_cod_t;

/* COC marker segment parameters. */

typedef struct {

        /* The component number. */
        uint_fast16_t compno;

        /* Component-related parameters. */
        jpc_coxcp_t compparms;

} jpc_coc_t;

/**************************************\
* RGN marker segment parameters.
\**************************************/

/* The maxshift ROI style. */
#define JPC_RGN_MAXSHIFT        0x00

typedef struct {

        /* The component to which the marker applies. */
        uint_fast16_t compno;

        /* The ROI style. */
        uint_fast8_t roisty;

        /* The ROI shift value. */
        uint_fast8_t roishift;

} jpc_rgn_t;

/**************************************\
* QCD/QCC marker segment parameters.
\**************************************/

/*
 * Quantization style constants.
 */

#define JPC_QCX_NOQNT   0 /* No quantization. */
#define JPC_QCX_SIQNT   1 /* Scalar quantization, implicit. */
#define JPC_QCX_SEQNT   2 /* Scalar quantization, explicit. */

/*
 * Stepsize manipulation macros.
 */

JAS_ATTRIBUTE_CONST
static inline unsigned JPC_QCX_GETEXPN(unsigned x)
{
        return x » 11;
}

JAS_ATTRIBUTE_CONST
static inline unsigned JPC_QCX_GETMANT(unsigned x)
{
        return x & 0x7ff;
}

JAS_ATTRIBUTE_CONST
static inline uint_fast16_t JPC_QCX_EXPN(unsigned x)
{
```

```
        assert(!(x & (~0x1f)));

        return (x & 0x1f) « 11;
}

JAS_ATTRIBUTE_CONST
static inline uint_fast16_t JPC_QCX_MANT(unsigned x)
{
        assert(!(x & (~0x7ff)));

        return x & 0x7ff;
}

/* Per component information. */

typedef struct {

        /* The quantization style. */
        uint_fast8_t qntsty;

        /* The number of step sizes. */
        int numstepsizes;

        /* The step sizes. */
        uint_fast16_t *stepsizes;

        /* The number of guard bits. */
        uint_fast8_t numguard;

} jpc_qcxcp_t;

/* QCC marker segment parameters. */

typedef struct {

        /* The component associated with this marker segment. */
        uint_fast16_t compno;

        /* The parameters. */
        jpc_qcxcp_t compparms;

} jpc_qcc_t;

/* QCD marker segment parameters. */

typedef struct {

        /* The parameters. */
        jpc_qcxcp_t compparms;

} jpc_qcd_t;

/**************************************\
* POD marker segment parameters.
\**************************************/

typedef struct {

        /* The progression order. */
        uint_fast8_t prgord;

        /* The lower bound (inclusive) on the resolution level for the
          progression order volume. */
        uint_fast8_t rlvlnostart;

        /* The upper bound (exclusive) on the resolution level for the
          progression order volume. */
        uint_fast8_t rlvlnoend;

        /* The lower bound (inclusive) on the component for the progression
          order volume. */
        uint_fast16_t compnostart;

        /* The upper bound (exclusive) on the component for the progression
          order volume. */
        uint_fast16_t compnoend;

        /* The upper bound (exclusive) on the layer for the progression
          order volume. */
        uint_fast16_t lyrnoend;

```

```
} jpc_pocpchg_t;

/* An alias for the above type. */
typedef jpc_pocpchg_t jpc_pchg_t;

/* POC marker segment parameters. */

typedef struct {

        /* The number of progression order changes. */
        int numpchgs;

        /* The per-progression-order-change information. */
        jpc_pocpchg_t *pchgs;

} jpc_poc_t;

/**************************************\
* PPM/PPT marker segment parameters.
\**************************************/

/* PPM marker segment parameters. */

typedef struct {

        /* The index. */
        uint_fast8_t ind;

        /* The length. */
        uint_fast16_t len;

        /* The data. */
        jas_uchar *data;

} jpc_ppm_t;

/* PPT marker segment parameters. */

typedef struct {

        /* The index. */
        uint_fast8_t ind;

        /* The length. */
        uint_fast32_t len;

        /* The data. */
        unsigned char *data;

} jpc_ppt_t;

/**************************************\
* COM marker segment parameters.
\**************************************/

/*
 * Registration IDs.
 */

#define JPC_COM_BIN             0x00
#define JPC_COM_LATIN   0x01

typedef struct {

        /* The registration ID. */
        uint_fast16_t regid;

        /* The length of the data in bytes. */
        uint_fast16_t len;

        /* The data. */
        jas_uchar *data;

} jpc_com_t;

/**************************************\
* SOP marker segment parameters.
\**************************************/

typedef struct {

```

```
        /* The sequence number. */
        uint_fast16_t seqno;

} jpc_sop_t;

/*************************************\
* CRG marker segment parameters.
\*************************************/

/* Per component information. */

typedef struct {

        /* The horizontal offset. */
        uint_fast16_t hoff;

        /* The vertical offset. */
        uint_fast16_t voff;

} jpc_crgcomp_t;

typedef struct {

        /* The number of components. */
        int numcomps;

        /* Per component information. */
        jpc_crgcomp_t *comps;

} jpc_crg_t;

/*************************************\
* Marker segment parameters for unknown marker type.
\*************************************/

typedef struct {

        /* The data. */
        jas_uchar *data;

        /* The length. */
        uint_fast16_t len;

} jpc_unk_t;

/*************************************\
* Generic marker segment parameters.
\*************************************/

typedef union {
        int soc;        /* unused */
        jpc_sot_t sot;
        int sod;        /* unused */
        int eoc;        /* unused */
        jpc_siz_t siz;
        jpc_cod_t cod;
        jpc_coc_t coc;
        jpc_rgn_t rgn;
        jpc_qcd_t qcd;
        jpc_qcc_t qcc;
        jpc_poc_t poc;
        /* jpc_plm_t plm; */
        /* jpc_plt_t plt; */
        jpc_ppm_t ppm;
        jpc_ppt_t ppt;
        jpc_sop_t sop;
        int eph;        /* unused */
        jpc_com_t com;
        jpc_crg_t crg;
        jpc_unk_t unk;
} jpc_msparms_t;

/*************************************\
* Marker segment.
\*************************************/

/* Marker segment IDs. */

/* The smallest valid marker value. */
#define JPC_MS_MIN      0xff00

```

```
/* The largest valid marker value. */
#define JPC_MS_MAX      0xffff

/* The minimum marker value that cannot occur within packet data. */
#define JPC_MS_INMIN    0xff80
/* The maximum marker value that cannot occur within packet data. */
#define JPC_MS_INMAX    0xffff

/* Delimiting marker segments. */
#define JPC_MS_SOC      0xff4f /* Start of code stream (SOC). */
#define JPC_MS_SOT      0xff90 /* Start of tile-part (SOT). */
#define JPC_MS_SOD      0xff93 /* Start of data (SOD). */
#define JPC_MS_EOC      0xffd9 /* End of code stream (EOC). */

/* Fixed information marker segments. */
#define JPC_MS_SIZ      0xff51 /* Image and tile size (SIZ). */

/* Functional marker segments. */
#define JPC_MS_COD      0xff52 /* Coding style default (COD). */
#define JPC_MS_COC      0xff53 /* Coding style component (COC). */
#define JPC_MS_RGN      0xff5e /* Region of interest (RGN). */
#define JPC_MS_QCD      0xff5c /* Quantization default (QCD). */
#define JPC_MS_QCC      0xff5d /* Quantization component (QCC). */
#define JPC_MS_POC      0xff5f /* Progression order default (POC). */

/* Pointer marker segments. */
#define JPC_MS_TLM      0xff55 /* Tile-part lengths, main header (TLM). */
#define JPC_MS_PLM      0xff57 /* Packet length, main header (PLM). */
#define JPC_MS_PLT      0xff58 /* Packet length, tile-part header (PLT). */
#define JPC_MS_PPM      0xff60 /* Packed packet headers, main header (PPM). */
#define JPC_MS_PPT      0xff61 /* Packet packet headers, tile-part header (PPT). */

/* In bit stream marker segments. */
#define JPC_MS_SOP      0xff91  /* Start of packet (SOP). */
#define JPC_MS_EPH      0xff92  /* End of packet header (EPH). */

/* Informational marker segments. */
#define JPC_MS_CRG      0xff63 /* Component registration (CRG). */
#define JPC_MS_COM      0xff64 /* Comment (COM). */

/* Forward declaration. */
struct jpc_msops_s;

/* Generic marker segment class. */

typedef struct {

        /* The type of marker segment. */
        uint_fast16_t id;

        /* The length of the marker segment. */
        uint_fast16_t len;

        /* The starting offset within the stream. */
        uint_fast32_t off;

        /* The parameters of the marker segment. */
        jpc_msparms_t parms;

        /* The marker segment operations. */
        const struct jpc_msops_s *ops;

} jpc_ms_t;

/* Marker segment operations (which depend on the marker segment type). */

typedef struct jpc_msops_s {

        /* Destroy the marker segment parameters. */
        void (*destroyparms)(jpc_ms_t *ms);

        /* Get the marker segment parameters from a stream. */
        int (*getparms)(jpc_ms_t *ms, jpc_cstate_t *cstate, jas_stream_t *in);

        /* Put the marker segment parameters to a stream. */
        int (*putparms)(jpc_ms_t *ms, jpc_cstate_t *cstate, jas_stream_t *out);

        /* Dump the marker segment parameters (for debugging). */
        int (*dumpparms)(jpc_ms_t *ms);

} jpc_msops_t;
```

```

/*****************************************************************************\
* Macros/Functions.
\*****************************************************************************/

/* Create a code-stream state object. */
jpc_cstate_t *jpc_cstate_create(void);

/* Destroy a code-stream state object. */
void jpc_cstate_destroy(jpc_cstate_t *cstate);

/* Create a marker segment. */
jpc_ms_t *jpc_ms_create(int type);

/* Destroy a marker segment. */
void jpc_ms_destroy(jpc_ms_t *ms);

/* Does a marker segment have parameters? */
JAS_ATTRIBUTE_CONST
static inline bool JPC_MS_HASPARMS(unsigned x)
{
        return !(x == JPC_MS_SOC || x == JPC_MS_SOD || x == JPC_MS_EOC ||
                 x == JPC_MS_EPH || (x >= 0xff30 && x <= 0xff3f));
}

/* Get the marker segment type. */
JAS_ATTRIBUTE_PURE
static inline unsigned jpc_ms_gettype(const jpc_ms_t *ms)
{
        return ms->id;
}

/* Read a marker segment from a stream. */
jpc_ms_t *jpc_getms(jas_stream_t *in, jpc_cstate_t *cstate);

/* Write a marker segment to a stream. */
int jpc_putms(jas_stream_t *out, jpc_cstate_t *cstate, jpc_ms_t *ms);

/* Copy code stream data from one stream to another. */
int jpc_getdata(jas_stream_t *in, jas_stream_t *out, long n);

/* Copy code stream data from one stream to another. */
int jpc_putdata(jas_stream_t *out, jas_stream_t *in, long n);

/* Dump a marker segment (for debugging). */
void jpc_ms_dump(jpc_ms_t *ms);

/* Read a 8-bit unsigned integer from a stream. */
int jpc_getuint8(jas_stream_t *in, uint_fast8_t *val);

/* Read a 16-bit unsigned integer from a stream. */
int jpc_getuint16(jas_stream_t *in, uint_fast16_t *val);

/* Read a 32-bit unsigned integer from a stream. */
int jpc_getuint32(jas_stream_t *in, uint_fast32_t *val);

/* Write a 8-bit unsigned integer to a stream. */
int jpc_putuint8(jas_stream_t *out, uint_fast8_t val);

/* Write a 16-bit unsigned integer to a stream. */
int jpc_putuint16(jas_stream_t *out, uint_fast16_t val);

/* Write a 32-bit unsigned integer to a stream. */
int jpc_putuint32(jas_stream_t *out, uint_fast32_t val);

#endif
```

16.50 jpc_dec.h

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

/* __START_OF_JASPER_LICENSE__
```

```
*
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * JPEG-2000 Decoder
 *
 * $Id$
 */

#ifndef JPC_DEC_H
#define JPC_DEC_H

/******************************************************************************\
* Includes.
\******************************************************************************/

#include "jasper/jas_stream.h"
#include "jasper/jas_image.h"

#include "jpc_cod.h"
#include "jpc_tsfb.h"
#include "jpc_tagtree.h"
#include "jpc_cs.h"
#include "jpc_t1cod.h"
#include "jpc_t2cod.h"

/******************************************************************************\
* Below are some ugly warts necessary to support packed packet headers.
\******************************************************************************/
```

```
/* PPM/PPT marker segment table entry. */

typedef struct {

        /* The index for this entry. */
        uint_fast16_t ind;

        /* The data length. */
        uint_fast32_t len;

         /* The data. */
         jas_uchar *data;

} jpc_ppxstabent_t;

/* PPM/PPT marker segment table. */

typedef struct {

        /* The number of entries. */
        unsigned numents;

        /* The maximum number of entries (i.e., the allocated size of the array
          below). */
        unsigned maxents;

        /* The table entries. */
        jpc_ppxstabent_t **ents;

} jpc_ppxstab_t;

/* Stream list class. */

typedef struct {

        /* The number of streams in this list. */
        unsigned numstreams;

        /* The maximum number of streams that can be accomodated without
          growing the streams array. */
        unsigned maxstreams;

        /* The streams. */
        jas_stream_t **streams;

} jpc_streamlist_t;

/*****************************************************************************\
* Coding parameters class.
\*****************************************************************************/

/* Per-component coding parameters. */

typedef struct {

        /* How were various coding parameters set? */
        unsigned flags;

        /* Per-component coding style parameters (e.g., explicit precinct sizes) */
        uint_fast8_t csty;

        /* The number of resolution levels. */
        uint_fast8_t numrlvls;

        /* The code block width exponent. */
        uint_fast8_t cblkwidthexpn;

        /* The code block height exponent. */
        uint_fast8_t cblkheightexpn;

        /* The QMFB ID. */
        uint_fast8_t qmfbid;

        /* The quantization style. */
        uint_fast8_t qsty;

        /* The number of quantizer step sizes. */
        uint_fast16_t numstepsizes;

        /* The step sizes. */
```

```
        uint_fast16_t stepsizes[3 * JPC_MAXRLVLS + 1];

        /* The number of guard bits. */
        uint_fast8_t numguardbits;

        /* The ROI shift value. */
        uint_fast8_t roishift;

        /* The code block parameters. */
        uint_fast8_t cblkctx;

        /* The precinct width exponents. */
        uint_fast8_t prcwidthexpns[JPC_MAXRLVLS];

        /* The precinct height exponents. */
        uint_fast8_t prcheightexpns[JPC_MAXRLVLS];

} jpc_dec_ccp_t;

/* Coding paramters. */

typedef struct {

        /* How were these coding parameters set? */
        unsigned flags;

        /* Progression change list. */
        jpc_pchglist_t *pchglist;

        /* Progression order. */
        uint_fast8_t prgord;

        /* The number of layers. */
        uint_fast16_t numlyrs;

        /* The MCT ID. */
        uint_fast8_t mctid;

        /* The coding style parameters (e.g., SOP, EPH). */
        uint_fast8_t csty;

        /* The number of components. */
        unsigned numcomps;

        /* The per-component coding parameters. */
        jpc_dec_ccp_t *ccps;

} jpc_dec_cp_t;

/******************************************************************************\
* Decoder class.
\******************************************************************************/

/* Decoder per-segment state information. */

typedef struct jpc_dec_seg_s {

        /* The next segment in the list. */
        struct jpc_dec_seg_s *next;

        /* The previous segment in the list. */
        struct jpc_dec_seg_s *prev;

        /* The starting pass number for this segment. */
        unsigned passno;

        /* The number of passes in this segment. */
        unsigned numpasses;

        /* The maximum number of passes in this segment. */
        unsigned maxpasses;

        /* The type of data in this segment (i.e., MQ or raw). */
        enum jpc_segtype type;

        /* A stream containing the data for this segment. */
        jas_stream_t *stream;

        /* The number of bytes destined for this segment from the packet
          currently being decoded. */
        unsigned cnt;
```

```
        /* A flag indicating if this segment has been terminated. */
        int complete;

        /* The layer number to which this segment belongs. */
        /* If the segment spans multiple layers, then the largest layer number
          spanned by the segment is used. */
        unsigned lyrno;

} jpc_dec_seg_t;

/* Decoder segment list. */

typedef struct {

        /* The first entry in the list. */
        jpc_dec_seg_t *head;

        /* The last entry in the list. */
        jpc_dec_seg_t *tail;

} jpc_dec_seglist_t;

/* Decoder per-code-block state information. */

typedef struct {

        /* The number of passes. */
        unsigned numpasses;

        /* A list of segments that still need to be decoded. */
        jpc_dec_seglist_t segs;

        /* The first incomplete/partial segment. */
        jpc_dec_seg_t *curseg;

        /* The number of leading insignificant bit planes for this code block. */
        unsigned numimsbs;

        /* The number of bits used to encode pass data lengths. */
        unsigned numlenbits;

        /* The first pass number containing data for this code block. */
        unsigned firstpassno;

        /* The sample data associated with this code block. */
        jas_matrix_t *data;

} jpc_dec_cblk_t;

/* Decoder per-code-block-group state information. */

typedef struct {

        /* The x-coordinate of the top-left corner of the precinct. */
        uint_fast32_t xstart;

        /* The y-coordinate of the top-left corner of the precinct. */
        uint_fast32_t ystart;

        /* The x-coordinate of the bottom-right corner of the precinct
          (plus one). */
        uint_fast32_t xend;

        /* The y-coordinate of the bottom-right corner of the precinct
          (plus one). */
        uint_fast32_t yend;

        /* The number of code blocks spanning this precinct in the horizontal
          direction. */
        unsigned numhcblks;

        /* The number of code blocks spanning this precinct in the vertical
          direction. */
        unsigned numvcblks;

        /* The total number of code blocks in this precinct. */
        unsigned numcblks;

        /* The per code block information. */
        jpc_dec_cblk_t *cblks;
```

```
        /* The inclusion tag tree. */
        jpc_tagtree_t *incltagtree;

        /* The insignificant MSBs tag tree. */
        jpc_tagtree_t *numimsbstagtree;

} jpc_dec_prc_t;

/* Decoder per-band state information. */

typedef struct {

        /* The per-code-block-group state information. */
        jpc_dec_prc_t *prcs;

        /* The sample data associated with this band. */
        jas_matrix_t *data;

        /* The orientation of this band (i.e., LL, LH, HL, or HH). */
        enum jpc_tsfb_orient orient;

        /* The encoded quantizer step size. */
        unsigned stepsize;

        /* The absolute quantizer step size. */
        jpc_fix_t absstepsize;

        /* The number of bit planes for this band. */
        unsigned numbps;

        /* The analysis gain associated with this band. */
        int analgain;

        /* The ROI shift value for this band. */
        int roishift;

} jpc_dec_band_t;

/* Decoder per-resolution-level state information. */

typedef struct {

        /* The number of bands associated with this resolution level. */
        unsigned numbands;

        /* The per-band information. */
        jpc_dec_band_t *bands;

        /* The x-coordinate of the top-left corner of the tile-component
          at this resolution. */
        uint_fast32_t xstart;

        /* The y-coordinate of the top-left corner of the tile-component
          at this resolution. */
        uint_fast32_t ystart;

        /* The x-coordinate of the bottom-right corner of the tile-component
          at this resolution (plus one). */
        uint_fast32_t xend;

        /* The y-coordinate of the bottom-right corner of the tile-component
          at this resolution (plus one). */
        uint_fast32_t yend;

        /* The exponent value for the nominal precinct width measured
          relative to the associated LL band. */
        unsigned prcwidthexpn;

        /* The exponent value for the nominal precinct height measured
          relative to the associated LL band. */
        unsigned prcheightexpn;

        /* The number of precincts in the horizontal direction. */
        unsigned numhprcs;

        /* The number of precincts in the vertical direction. */
        unsigned numvprcs;

        /* The total number of precincts. */
        unsigned numprcs;
```

```
        /* The exponent value for the nominal code block group width.
          This quantity is associated with the next lower resolution level
          (assuming that there is one). */
        unsigned cbgwidthexpn;

        /* The exponent value for the nominal code block group height
          This quantity is associated with the next lower resolution level
          (assuming that there is one). */
        unsigned cbgheightexpn;

        /* The exponent value for the code block width. */
        uint_fast16_t cblkwidthexpn;

        /* The exponent value for the code block height. */
        uint_fast16_t cblkheightexpn;

} jpc_dec_rlvl_t;

/* Decoder per-tile-component state information. */

typedef struct {

        /* The x-coordinate of the top-left corner of the tile-component
          in the coordinate system of the tile-component. */
        uint_fast32_t xstart;

        /* The y-coordinate of the top-left corner of the tile-component
          in the coordinate system of the tile-component. */
        uint_fast32_t ystart;

        /* The x-coordinate of the bottom-right corner of the tile-component
          in the coordinate system of the tile-component (plus one). */
        uint_fast32_t xend;

        /* The y-coordinate of the bottom-right corner of the tile-component
          in the coordinate system of the tile-component (plus one). */
        uint_fast32_t yend;

        /* The component data for the current tile. */
        jas_matrix_t *data;

        /* The number of resolution levels. */
        unsigned numrlvls;

        /* The per resolution level information. */
        jpc_dec_rlvl_t *rlvls;

        /* The TSFB. */
        jpc_tsfb_t *tsfb;

} jpc_dec_tcomp_t;

/*
 * Tile states.
 */

#define JPC_TILE_INIT   0
#define JPC_TILE_ACTIVE 1
#define JPC_TILE_ACTIVELAST     2
#define JPC_TILE_DONE   3

/* Decoder per-tile state information. */

typedef struct {

        /* The processing state for this tile. */
        int state;

        /* The x-coordinate of the top-left corner of the tile on the reference
          grid. */
        uint_fast32_t xstart;

        /* The y-coordinate of the top-left corner of the tile on the reference
          grid. */
        uint_fast32_t ystart;

        /* The x-coordinate of the bottom-right corner of the tile on the
          reference grid (plus one). */
        uint_fast32_t xend;

```

```
        /* The y-coordinate of the bottom-right corner of the tile on the
          reference grid (plus one). */
        uint_fast32_t yend;

        /* The packed packet header data for this tile. */
        jpc_ppxstab_t *pptstab;

        /* A stream containing the packed packet header data for this tile. */
        jas_stream_t *pkthdrstream;

        /* The coding parameters for this tile. */
        jpc_dec_cp_t *cp;

        /* The per tile-component information. */
        jpc_dec_tcomp_t *tcomps;

        /* The next expected tile-part number. */
        unsigned partno;

        /* The number of tile-parts. */
        unsigned numparts;

        /* The coding mode. */
        int realmode;

        /* The packet iterator for this tile. */
        jpc_pi_t *pi;

} jpc_dec_tile_t;

/* Decoder per-component state information. */

typedef struct {

        /* The horizontal sampling period. */
        uint_fast32_t hstep;

        /* The vertical sampling period. */
        uint_fast32_t vstep;

        /* The number of samples in the horizontal direction. */
        uint_fast32_t width;

        /* The number of samples in the vertical direction. */
        uint_fast32_t height;

        /* The precision of the sample data. */
        uint_fast16_t prec;

        /* The signedness of the sample data. */
        bool sgnd;

        /* The sample alignment horizontal offset. */
        uint_fast32_t hsubstep;

        /* The sample alignment vertical offset. */
        uint_fast32_t vsubstep;

} jpc_dec_cmpt_t;

/* Decoder state information. */

typedef struct {

        /* The decoded image. */
        jas_image_t *image;

        /* The x-coordinate of the top-left corner of the image area on
          the reference grid. */
        uint_fast32_t xstart;

        /* The y-coordinate of the top-left corner of the image area on
          the reference grid. */
        uint_fast32_t ystart;

        /* The x-coordinate of the bottom-right corner of the image area on
          the reference grid (plus one). */
        uint_fast32_t xend;

        /* The y-coordinate of the bottom-right corner of the image area on
          the reference grid (plus one). */
```

```
        uint_fast32_t yend;

        /* The nominal tile width in units of the image reference grid. */
        uint_fast32_t tilewidth;

        /* The nominal tile height in units of the image reference grid. */
        uint_fast32_t tileheight;

        /* The horizontal offset from the origin of the reference grid to the
          left side of the first tile. */
        uint_fast32_t tilexoff;

        /* The vertical offset from the origin of the reference grid to the
          top side of the first tile. */
        uint_fast32_t tileyoff;

        /* The number of tiles spanning the image area in the vertical
          direction. */
        unsigned numhtiles;

        /* The number of tiles spanning the image area in the horizontal
          direction. */
        unsigned numvtiles;

        /* The total number of tiles. */
        unsigned numtiles;

        /* The per-tile information. */
        jpc_dec_tile_t *tiles;

        /* The tile currently being processed. */
        jpc_dec_tile_t *curtile;

        /* The number of components. */
        unsigned numcomps;

        /* The stream containing the input JPEG-2000 code stream data. */
        jas_stream_t *in;

        /* The default coding parameters for all tiles. */
        jpc_dec_cp_t *cp;

        /* The maximum number of layers that may be decoded. */
        unsigned maxlyrs;

        /* The maximum number of packets that may be decoded. */
        int maxpkts;

        /* The number of packets decoded so far in the processing of the entire
          code stream. */
        unsigned numpkts;

        /* The next expected PPM marker segment sequence number. */
        unsigned ppmseqno;

        /* The current state for code stream processing. */
        int state;

        /* The per-component information. */
        jpc_dec_cmpt_t *cmpts;

        /* The information from PPM marker segments. */
        jpc_ppxstab_t *ppmstab;

        /* A list of streams containing packet header data from PPM marker
          segments. */
        jpc_streamlist_t *pkthdrstreams;

        /* The expected ending offset for a tile-part. */
        long curtileendoff;

        /* This is required by the tier-2 decoder. */
        jpc_cstate_t *cstate;

        size_t max_samples;

} jpc_dec_t;

/* Decoder options. */

typedef struct {
```

```

        /* The debug level for the decoder. */
        int debug;

        /* The maximum number of layers to decode. */
        unsigned maxlyrs;

        /* The maximum number of packets to decode. */
        int maxpkts;

        size_t max_samples;

} jpc_dec_importopts_t;

/*****************************************************************************\
* Functions.
\*****************************************************************************/

/* Create a decoder segment object. */
jpc_dec_seg_t *jpc_seg_alloc(void);

/* Destroy a decoder segment object. */
void jpc_seg_destroy(jpc_dec_seg_t *seg);

/* Remove a segment from a segment list. */
void jpc_seglist_remove(jpc_dec_seglist_t *list, jpc_dec_seg_t *node);

/* Insert a segment into a segment list. */
void jpc_seglist_insert(jpc_dec_seglist_t *list, jpc_dec_seg_t *ins,
  jpc_dec_seg_t *node);

#endif
```

16.51 jpc_enc.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * $Id$
 */

#ifndef JPC_ENC_H
#define JPC_ENC_H

/*****************************************************************************\
* Includes.
\*****************************************************************************/

#include "jasper/jas_image.h"
#include "jasper/jas_seq.h"
#include "jasper/jas_stream.h"

#include "jpc_cod.h"
#include "jpc_t1cod.h"
#include "jpc_t2cod.h"
#include "jpc_mqenc.h"
#include "jpc_tagtree.h"
#include "jpc_cs.h"
#include "jpc_fix.h"
#include "jpc_flt.h"
#include "jpc_tsfb.h"

/*****************************************************************************\
* Constants.
\*****************************************************************************/

/* The number of bits used in various lookup tables. */
#define JPC_NUMEXTRABITS        JPC_NMSEDEC_FRACBITS

/* An invalid R-D slope value. */
#define JPC_BADRDSLOPE  (-1)

/*****************************************************************************\
* Coding parameters types.
\*****************************************************************************/

/* Per-component coding paramters. */

typedef struct {

        /* The horizontal sampling period. */
        uint_fast8_t sampgrdstepx;

        /* The vertical sampling period. */
        uint_fast8_t sampgrdstepy;

        /* The sample alignment horizontal offset. */
        uint_fast8_t sampgrdsubstepx;

        /* The sample alignment vertical offset. */
        uint_fast8_t sampgrdsubstepy;

        /* The precision of the samples. */
        uint_fast8_t prec;

        /* The signedness of the samples. */
        bool sgnd;
```

```

        /* The number of step sizes. */
        uint_fast16_t numstepsizes;

        /* The quantizer step sizes. */
        uint_fast16_t stepsizes[JPC_MAXBANDS];

} jpc_enc_ccp_t;

/* Per-tile coding parameters. */

typedef struct {

        /* The coding mode. */
        bool intmode;

        /* The coding style (i.e., SOP, EPH). */
        uint_fast8_t csty;

        /* The progression order. */
        uint_fast8_t prg;

        /* The multicomponent transform. */
        uint_fast8_t mctid;

        /* The number of layers. */
        uint_fast16_t numlyrs;

        /* The normalized bit rates associated with the various
          intermediate layers. */
        jpc_fix_t *ilyrrates;

} jpc_enc_tcp_t;

/* Per tile-component coding parameters. */

typedef struct {

        /* The coding style (i.e., explicit precinct sizes). */
        uint_fast8_t csty;

        /* The maximum number of resolution levels allowed. */
        uint_fast8_t maxrlvls;

        /* The exponent for the nominal code block width. */
        uint_fast16_t cblkwidthexpn;

        /* The exponent for the nominal code block height. */
        uint_fast16_t cblkheightexpn;

        /* The code block style parameters (e.g., lazy, terminate all,
          segmentation symbols, causal, reset probability models). */
        uint_fast8_t cblksty;

        /* The QMFB. */
        uint_fast8_t qmfbid;

        /* The precinct width values. */
        uint_fast16_t prcwidthexpns[JPC_MAXRLVLS];

        /* The precinct height values. */
        uint_fast16_t prcheightexpns[JPC_MAXRLVLS];

        /* The number of guard bits. */
        uint_fast8_t numgbits;

} jpc_enc_tccp_t;

/* Coding parameters. */

typedef struct {

        /* The debug level. */
        int debug;

        /* The horizontal offset from the origin of the reference grid to the
          left edge of the image area. */
        uint_fast32_t imgareatlx;

        /* The vertical offset from the origin of the reference grid to the
          top edge of the image area. */
```

```
        uint_fast32_t imgareatly;

        /* The horizontal offset from the origin of the reference grid to the
          right edge of the image area (plus one). */
        uint_fast32_t refgrdwidth;

        /* The vertical offset from the origin of the reference grid to the
          bottom edge of the image area (plus one). */
        uint_fast32_t refgrdheight;

        /* The horizontal offset from the origin of the tile grid to the
          origin of the reference grid. */
        uint_fast32_t tilegrdoffx;

        /* The vertical offset from the origin of the tile grid to the
          origin of the reference grid. */
        uint_fast32_t tilegrdoffy;

        /* The nominal tile width in units of the image reference grid. */
        uint_fast32_t tilewidth;

        /* The nominal tile height in units of the image reference grid. */
        uint_fast32_t tileheight;

        /* The number of tiles spanning the image area in the horizontal
          direction. */
        uint_fast32_t numhtiles;

        /* The number of tiles spanning the image area in the vertical
          direction. */
        uint_fast32_t numvtiles;

        /* The number of tiles. */
        uint_fast32_t numtiles;

        /* The number of components. */
        uint_fast16_t numcmpts;

        /* The per-component coding parameters. */
        jpc_enc_ccp_t *ccps;

        /* The per-tile coding parameters. */
        jpc_enc_tcp_t tcp;

        /* The per-tile-component coding parameters. */
        jpc_enc_tccp_t tccp;

        /* The target code stream length in bytes. */
        uint_fast32_t totalsize;

        /* The raw (i.e., uncompressed) size of the image in bytes. */
        uint_fast32_t rawsize;

} jpc_enc_cp_t;

/*******************************************************************************\
* Encoder class.
\*******************************************************************************/

/* Encoder per-coding-pass state information. */

typedef struct {

        /* The starting offset for this pass. */
        int start;

        /* The ending offset for this pass. */
        int end;

        /* The type of data in this pass (i.e., MQ or raw). */
        enum jpc_segtype type;

        /* Flag indicating that this pass is terminated. */
        int term;

        /* The entropy coder state after coding this pass. */
        jpc_mqencstate_t mqencstate;

        /* The layer to which this pass has been assigned. */
        unsigned lyrno;

```

```
        /* The R-D slope for this pass. */
        jpc_flt_t rdslope;

        /* The weighted MSE reduction associated with this pass. */
        jpc_flt_t wmsedec;

        /* The cumulative weighted MSE reduction. */
        jpc_flt_t cumwmsedec;

        /* The normalized MSE reduction. */
        long nmsedec;

} jpc_enc_pass_t;

/* Encoder per-code-block state information. */

typedef struct {

        /* The number of passes. */
        unsigned numpasses;

        /* The per-pass information. */
        jpc_enc_pass_t *passes;

        /* The number of passes encoded so far. */
        int numencpasses;

        /* The number of insignificant MSBs. */
        int numimsbs;

        /* The number of bits used to encode pass data lengths. */
        int numlenbits;

        /* The byte stream for this code block. */
        jas_stream_t *stream;

        /* The entropy encoder. */
        jpc_mqenc_t *mqenc;

        /* The data for this code block. */
        jas_matrix_t *data;

        /* The state for this code block. */
        jas_matrix_t *flags;

        /* The number of bit planes required for this code block. */
        int numbps;

        /* The next pass to be encoded. */
        jpc_enc_pass_t *curpass;

        /* The per-code-block-group state information. */
        struct jpc_enc_prc_s *prc;

        /* The saved current pass. */
        /* This is used by the rate control code. */
        jpc_enc_pass_t *savedcurpass;

        /* The saved length indicator size. */
        /* This is used by the rate control code. */
        int savednumlenbits;

        /* The saved number of encoded passes. */
        /* This is used by the rate control code. */
        int savednumencpasses;

} jpc_enc_cblk_t;

/* Encoder per-code-block-group state information. */

typedef struct jpc_enc_prc_s {

        /* The x-coordinate of the top-left corner of the precinct. */
        uint_fast32_t tlx;

        /* The y-coordinate of the top-left corner of the precinct. */
        uint_fast32_t tly;

        /* The x-coordinate of the bottom-right corner of the precinct
          (plus one). */
        uint_fast32_t brx;
```

```
        /* The y-coordinate of the bottom-right corner of the precinct
          (plus one). */
        uint_fast32_t bry;

        /* The number of code blocks spanning the precinct in the horizontal
        direction. */
        int numhcblks;

        /* The number of code blocks spanning the precinct in the vertical
        direction. */
        int numvcblks;

        /* The total number of code blocks. */
        unsigned numcblks;

        /* The per-code-block information. */
        jpc_enc_cblk_t *cblks;

        /* The inclusion tag tree. */
        jpc_tagtree_t *incltree;

        /* The insignifcant MSBs tag tree. */
        jpc_tagtree_t *nlibtree;

        /* The per-band information. */
        struct jpc_enc_band_s *band;

        /* The saved inclusion tag tree. */
        /* This is used by rate control. */
        jpc_tagtree_t *savincltree;

        /* The saved leading-insignificant-bit-planes tag tree. */
        /* This is used by rate control. */
        jpc_tagtree_t *savnlibtree;

} jpc_enc_prc_t;

/* Encoder per-band state information. */

typedef struct jpc_enc_band_s {

        /* The per precinct information. */
        jpc_enc_prc_t *prcs;

        /* The coefficient data for this band. */
        jas_matrix_t *data;

        /* The orientation of this band (i.e., LL, LH, HL, or HH). */
        enum jpc_tsfb_orient orient;

        /* The number of bit planes associated with this band. */
        int numbps;

        /* The quantizer step size. */
        jpc_fix_t absstepsize;

        /* The encoded quantizer step size. */
        int stepsize;

        /* The L2 norm of the synthesis basis functions associated with
          this band.  (The MCT is not considered in this value.) */
        jpc_fix_t synweight;

        /* The analysis gain for this band. */
        int analgain;

        /* The per-resolution-level information. */
        struct jpc_enc_rlvl_s *rlvl;

} jpc_enc_band_t;

/* Encoder per-resolution-level state information. */

typedef struct jpc_enc_rlvl_s {

        /* The x-coordinate of the top-left corner of the tile-component
          at this resolution. */
        uint_fast32_t tlx;

        /* The y-coordinate of the top-left corner of the tile-component
```

```
          at this resolution. */
        uint_fast32_t tly;

        /* The x-coordinate of the bottom-right corner of the tile-component
          at this resolution (plus one). */
        uint_fast32_t brx;

        /* The y-coordinate of the bottom-right corner of the tile-component
          at this resolution (plus one). */
        uint_fast32_t bry;

        /* The exponent value for the nominal precinct width measured
          relative to the associated LL band. */
        int prcwidthexpn;

        /* The exponent value for the nominal precinct height measured
          relative to the associated LL band. */
        int prcheightexpn;

        /* The number of precincts spanning the resolution level in the
          horizontal direction. */
        int numhprcs;

        /* The number of precincts spanning the resolution level in the
          vertical direction. */
        int numvprcs;

        /* The total number of precincts. */
        unsigned numprcs;

        /* The exponent value for the nominal code block group width.
          This quantity is associated with the next lower resolution level
          (assuming that there is one). */
        unsigned cbgwidthexpn;

        /* The exponent value for the nominal code block group height.
          This quantity is associated with the next lower resolution level
          (assuming that there is one). */
        unsigned cbgheightexpn;

        /* The exponent value for the code block width. */
        uint_fast16_t cblkwidthexpn;

        /* The exponent value for the code block height. */
        uint_fast16_t cblkheightexpn;

        /* The number of bands associated with this resolution level. */
        unsigned numbands;

        /* The per-band information. */
        jpc_enc_band_t *bands;

        /* The parent tile-component. */
        struct jpc_enc_tcmpt_s *tcmpt;

} jpc_enc_rlvl_t;

/* Encoder per-tile-component state information. */

typedef struct jpc_enc_tcmpt_s {

        /* The number of resolution levels. */
        unsigned numrlvls;

        /* The per-resolution-level information. */
        jpc_enc_rlvl_t *rlvls;

        /* The tile-component data. */
        jas_matrix_t *data;

        /* The QMFB. */
        int qmfbid;

        /* The number of bands. */
        int numbands;

        /* The TSFB. */
        jpc_tsfb_t *tsfb;

        /* The synthesis energy weight (for the MCT). */
        jpc_fix_t synweight;
```

```
        /* The precinct width exponents. */
        int prcwidthexpns[JPC_MAXRLVLS];

        /* The precinct height exponents. */
        int prcheightexpns[JPC_MAXRLVLS];

        /* The code block width exponent. */
        int cblkwidthexpn;

        /* The code block height exponent. */
        int cblkheightexpn;

        /* Coding style (i.e., explicit precinct sizes). */
        int csty;

        /* Code block style. */
        int cblksty;

        /* The number of quantizer step sizes. */
        int numstepsizes;

        /* The encoded quantizer step sizes. */
        uint_fast16_t stepsizes[JPC_MAXBANDS];

        /* The parent tile. */
        struct jpc_enc_tile_s *tile;

} jpc_enc_tcmpt_t;

/* Encoder per-tile state information. */

typedef struct jpc_enc_tile_s {

        /* The tile number. */
        uint_fast32_t tileno;

        /* The x-coordinate of the top-left corner of the tile measured with
          respect to the reference grid. */
        uint_fast32_t tlx;

        /* The y-coordinate of the top-left corner of the tile measured with
          respect to the reference grid. */
        uint_fast32_t tly;

        /* The x-coordinate of the bottom-right corner of the tile measured
          with respect to the reference grid (plus one). */
        uint_fast32_t brx;

        /* The y-coordinate of the bottom-right corner of the tile measured
          with respect to the reference grid (plus one). */
        uint_fast32_t bry;

        /* The coding style. */
        uint_fast8_t csty;

        /* The progression order. */
        uint_fast8_t prg;

        /* The number of layers. */
        unsigned numlyrs;

        /* The MCT to employ (if any). */
        uint_fast8_t mctid;

        /* The packet iterator (used to determine the order of packet
          generation). */
        jpc_pi_t *pi;

        /* The coding mode (i.e., integer or real). */
        bool intmode;

        /* The number of bytes to allocate to the various layers. */
        uint_fast32_t *lyrsizes;

        /* The number of tile-components. */
        unsigned numtcmpts;

        /* The per tile-component information. */
        jpc_enc_tcmpt_t *tcmpts;

```

```
        /* The raw (i.e., uncompressed) size of this tile. */
        uint_fast32_t rawsize;

} jpc_enc_tile_t;

/* Encoder class. */

typedef struct jpc_enc_s {

        /* The image being encoded. */
        jas_image_t *image;

        /* The output stream. */
        jas_stream_t *out;

        /* The coding parameters. */
        jpc_enc_cp_t *cp;

        /* The tile currently being processed. */
        jpc_enc_tile_t *curtile;

        /* The code stream state. */
        jpc_cstate_t *cstate;

        /* The number of bytes output so far. */
        uint_fast32_t len;

        /* The number of bytes available for the main body of the code stream. */
        /* This is used for rate allocation purposes. */
        uint_fast32_t mainbodysize;

        /* The marker segment currently being processed. */
        /* This member is a convenience for making cleanup easier. */
        jpc_ms_t *mrk;

        /* The stream used to temporarily hold tile-part data. */
        jas_stream_t *tmpstream;

} jpc_enc_t;

#endif
```

16.52 jpc_fix.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * Fixed-Point Number Class
 *
 * $Id$
 */

#ifndef JPC_FIX_H
#define JPC_FIX_H

/******************************************************************************\
* Includes.
\******************************************************************************/

#include "jasper/jas_config.h"
#include "jasper/jas_types.h"
#include "jasper/jas_fix.h"
#include "jasper/jas_math.h"

/******************************************************************************\
* Basic parameters of the fixed-point type.
\******************************************************************************/

/* The integral type used to represent a fixed-point number.  This
  type must be capable of representing values from -(2^31) to 2^31-1
  (inclusive). */
#ifdef JAS_ENABLE_32BIT
typedef int_least32_t jpc_fix_t;
#else
typedef int_fast32_t jpc_fix_t;
#endif

/* The integral type used to respresent higher-precision intermediate results.
  This type should be capable of representing values from -(2^63) to 2^63-1
  (inclusive). */
typedef int_fast64_t jpc_fix_big_t;

/* The number of bits used for the fractional part of a fixed-point number. */
#define JPC_FIX_FRACBITS        13

/******************************************************************************\
* Instantiations of the generic fixed-point number macros for the
* parameters given above.  (Too bad C does not support templates, eh?)
* The purpose of these macros is self-evident if one examines the
* corresponding macros in the jasper/jas_fix.h header file.
\******************************************************************************/

#define JPC_FIX_ZERO    JAS_FIX_ZERO(jpc_fix_t, JPC_FIX_FRACBITS)
#define JPC_FIX_ONE             JAS_FIX_ONE(jpc_fix_t, JPC_FIX_FRACBITS)
#define JPC_FIX_HALF    JAS_FIX_HALF(jpc_fix_t, JPC_FIX_FRACBITS)

JAS_ATTRIBUTE_CONST
```

```
static inline jpc_fix_t jpc_inttofix(int x)
{
        return JAS_INTTOFIX(jpc_fix_t, JPC_FIX_FRACBITS, x);
}

JAS_ATTRIBUTE_CONST
static inline int jpc_fixtoint(jpc_fix_t x)
{
        return JAS_FIXTOINT(jpc_fix_t, JPC_FIX_FRACBITS, x);
}

JAS_ATTRIBUTE_CONST
static inline double jpc_fixtodbl(jpc_fix_t x)
{
        return JAS_FIXTODBL(jpc_fix_t, JPC_FIX_FRACBITS, x);
}

JAS_ATTRIBUTE_CONST
static inline jpc_fix_t jpc_dbltofix(double x)
{
        return JAS_DBLTOFIX(jpc_fix_t, JPC_FIX_FRACBITS, x);
}

JAS_ATTRIBUTE_CONST
static inline jpc_fix_t jpc_fix_add(jpc_fix_t x, jpc_fix_t y)
{
        return JAS_FIX_ADD(jpc_fix_t, JPC_FIX_FRACBITS, x, y);
}

JAS_ATTRIBUTE_CONST
static inline jpc_fix_t jpc_fix_sub(jpc_fix_t x, jpc_fix_t y)
{
        return JAS_FIX_SUB(jpc_fix_t, JPC_FIX_FRACBITS, x, y);
}

JAS_ATTRIBUTE_CONST
static inline jpc_fix_t jpc_fix_mul(jpc_fix_big_t x, jpc_fix_big_t y)
{
        return JAS_FIX_MUL(jpc_fix_t, JPC_FIX_FRACBITS, jpc_fix_big_t, x, y);
}

JAS_ATTRIBUTE_CONST
static inline jpc_fix_big_t jpc_fix_mulbyint(jpc_fix_big_t x, int y)
{
        return JAS_FIX_MUL(jpc_fix_t, JPC_FIX_FRACBITS, jpc_fix_big_t, x, y);
}

JAS_ATTRIBUTE_CONST
static inline jpc_fix_t jpc_fix_div(jpc_fix_big_t x, jpc_fix_t y)
{
        return JAS_FIX_DIV(jpc_fix_t, JPC_FIX_FRACBITS, jpc_fix_big_t, x, y);
}

JAS_ATTRIBUTE_CONST
static inline jpc_fix_t jpc_fix_neg(jpc_fix_t x)
{
        return JAS_FIX_NEG(jpc_fix_t, JPC_FIX_FRACBITS, x);
}

//#define       jpc_fix_asl(x, n)       JAS_FIX_ASL(jpc_fix_t, JPC_FIX_FRACBITS, x, n)
//#define       jpc_fix_asr(x, n)       JAS_FIX_ASR(jpc_fix_t, JPC_FIX_FRACBITS, x, n)

#ifdef JAS_ENABLE_32BIT
#define jpc_fix_asl jas_least32_asl
#define jpc_fix_asr jas_least32_asr
#else
#define jpc_fix_asl jas_fast32_asl
#define jpc_fix_asr jas_fast32_asr
#endif

#define jpc_fix_pluseq(x, y)    JAS_FIX_PLUSEQ(jpc_fix_t, JPC_FIX_FRACBITS, x, y)
#define jpc_fix_minuseq(x, y)   JAS_FIX_MINUSEQ(jpc_fix_t, JPC_FIX_FRACBITS, x, y)
#define jpc_fix_muleq(x, y)     \
        JAS_FIX_MULEQ(jpc_fix_t, JPC_FIX_FRACBITS, jpc_fix_big_t, x, y)

JAS_ATTRIBUTE_CONST
static inline jpc_fix_t jpc_fix_abs(jpc_fix_t x)
{
        return JAS_FIX_ABS(jpc_fix_t, JPC_FIX_FRACBITS, x);
}
```

```
JAS_ATTRIBUTE_CONST
static inline bool jpc_fix_isint(jpc_fix_t x)
{
        return JAS_FIX_ISINT(jpc_fix_t, JPC_FIX_FRACBITS, x);
}

JAS_ATTRIBUTE_CONST
static inline int jpc_fix_sgn(jpc_fix_t x)
{
        return JAS_FIX_SGN(jpc_fix_t, JPC_FIX_FRACBITS, x);
}

JAS_ATTRIBUTE_CONST
static inline jpc_fix_t jpc_fix_round(jpc_fix_t x)
{
        return JAS_FIX_ROUND(jpc_fix_t, JPC_FIX_FRACBITS, x);
}

JAS_ATTRIBUTE_CONST
static inline jpc_fix_t jpc_fix_floor(jpc_fix_t x)
{
        return JAS_FIX_FLOOR(jpc_fix_t, JPC_FIX_FRACBITS, x);
}

/*****************************************************************************\
* Extra macros for convenience.
\*****************************************************************************/

/* Compute the sum of three fixed-point numbers. */
JAS_ATTRIBUTE_CONST
static inline jpc_fix_t jpc_fix_add3(jpc_fix_t x, jpc_fix_t y, jpc_fix_t z)
{
        return jpc_fix_add(jpc_fix_add(x, y), z);
}

#endif
```

16.53 jpc_flt.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * Floating-Point Class
 *
 * $Id$
 */

#ifndef JPC_FLT_H
#define JPC_FLT_H

/* The code ought to be modified so this type is not used at all. */
/* Very few places in the code rely on floating-point arithmetic, aside
  from conversions in printf's. */
typedef double jpc_flt_t;

#endif
```

16.54 jpc_math.h

```
/*
 * Copyright (c) 2001-2002 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JPC_MATH_H
#define JPC_MATH_H

/*****************************************************************************\
* Includes
\*****************************************************************************/

#include "jasper/jas_config.h"

#include "jpc_fix.h"

/*****************************************************************************\
* Macros
\*****************************************************************************/

/* Compute the floor of the quotient of two integers. */
#define JPC_FLOORDIV(x, y)      ((x) / (y))

/* Compute the ceiling of the quotient of two integers. */
#define JPC_CEILDIV(x, y)       (((x) + (y) - 1) / (y))

/* Compute the floor of (x / 2^y). */
#define JPC_FLOORDIVPOW2(x, y)  ((x) » (y))

/* Compute the ceiling of (x / 2^y). */
#define JPC_CEILDIVPOW2(x, y)   (((x) + (1 « (y)) - 1) » (y))

/*****************************************************************************\
* Functions.
\*****************************************************************************/

/* Calculate the bit position of the first leading one in a nonnegative
  integer. */
JAS_ATTRIBUTE_CONST
int jpc_int_firstone(int x);

JAS_ATTRIBUTE_CONST
int jpc_fix_firstone(jpc_fix_t x);

/* Calculate the integer quantity floor(log2(x)), where x is a positive
  integer. */
JAS_ATTRIBUTE_CONST
unsigned jpc_floorlog2(uint_fast32_t x);

#endif
```

16.55 jpc_mct.h

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

```
 *   British Columbia.
 * Copyright (c) 2001-2002 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * Multicomponent Transform Code
 *
 * $Id$
 */

#ifndef JPC_MCT_H
#define JPC_MCT_H

/******************************************************************************\
* Includes.
\******************************************************************************/

#include "jpc_fix.h"

#include "jasper/jas_seq.h"

/******************************************************************************\
* Constants.
\******************************************************************************/
```

```

/*
 * Multicomponent transform IDs.
 */

#define JPC_MCT_NONE    0
#define JPC_MCT_ICT             1
#define JPC_MCT_RCT             2

/*****************************************************************************\
* Functions.
\*****************************************************************************/

/* Calculate the forward RCT. */
void jpc_rct(jas_matrix_t *c0, jas_matrix_t *c1, jas_matrix_t *c2);

/* Calculate the inverse RCT. */
void jpc_irct(jas_matrix_t *c0, jas_matrix_t *c1, jas_matrix_t *c2);

/* Calculate the forward ICT. */
void jpc_ict(jas_matrix_t *c0, jas_matrix_t *c1, jas_matrix_t *c2);

/* Calculate the inverse ICT. */
void jpc_iict(jas_matrix_t *c0, jas_matrix_t *c1, jas_matrix_t *c2);

#endif
```

16.56 jpc_mqcod.h

```
/*
 * Copyright (c) 1999-2000 Image Power, Inc. and the University of
 *   British Columbia.
 * Copyright (c) 2001-2003 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * MQ Arithmetic Coder
 *
 * $Id$
 */

#ifndef JPC_MQCOD_H
#define JPC_MQCOD_H

/******************************************************************************\
* Includes.
\******************************************************************************/

#include "jasper/jas_types.h"

/******************************************************************************\
* Types.
\******************************************************************************/

/*
 * MQ coder context information.
 */

typedef struct {

        /* The most probable symbol (MPS). */
        bool mps;

        /* The state index. */
        int_least8_t ind;

} jpc_mqctx_t;

/*
 * MQ coder state table entry.
 */

typedef struct jpc_mqstate_s {

        /* The Qe value. */
        uint_least16_t qeval;

        /* The MPS. */
        bool mps;

        /* The NMPS state. */
        const struct jpc_mqstate_s *nmps;

        /* The NLPS state. */
        const struct jpc_mqstate_s *nlps;

} jpc_mqstate_t;

/******************************************************************************\
* Data.
\******************************************************************************/

/* The state table for the MQ coder. */
extern const jpc_mqstate_t jpc_mqstates[];

#endif
```

16.57 jpc_mqdec.h

```
/*
 * Copyright (c) 1999-2000 Image Power, Inc. and the University of
 *   British Columbia.
 * Copyright (c) 2001-2003 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * MQ Arithmetic Decoder
 *
 * $Id$
 */

#ifndef JPC_MQDEC_H
#define JPC_MQDEC_H

/******************************************************************************\
* Includes.
\******************************************************************************/

#include "jasper/jas_types.h"
#include "jasper/jas_stream.h"
```

```

#include "jpc_mqcod.h"

#include <stdio.h>

/******************************************************************************\
* Types.
\******************************************************************************/

/* MQ arithmetic decoder. */

typedef struct {

        /* The C register. */
        uint_least32_t creg;

        /* The A register. */
        uint_least32_t areg;

        /* The CT register. */
        uint_least32_t ctreg;

         /* The current context. */
         const jpc_mqstate_t **curctx;

         /* The per-context information. */
         const jpc_mqstate_t **ctxs;

         /* The maximum number of contexts. */
         unsigned maxctxs;

         /* The stream from which to read data. */
         jas_stream_t *in;

         /* The last character read. */
         jas_uchar inbuffer;

         /* The EOF indicator. */
         bool eof;

} jpc_mqdec_t;

/******************************************************************************\
* Functions/macros for construction and destruction.
\******************************************************************************/

/* Create a MQ decoder. */
jpc_mqdec_t *jpc_mqdec_create(unsigned maxctxs, jas_stream_t *in);

/* Destroy a MQ decoder. */
void jpc_mqdec_destroy(jpc_mqdec_t *dec);

/******************************************************************************\
* Functions/macros for initialization.
\******************************************************************************/

/* Set the input stream associated with a MQ decoder. */
void jpc_mqdec_setinput(jpc_mqdec_t *dec, jas_stream_t *in);

/* Initialize a MQ decoder. */
void jpc_mqdec_init(jpc_mqdec_t *dec);

/******************************************************************************\
* Functions/macros for manipulating contexts.
\******************************************************************************/

/* Set the current context for a MQ decoder. */
static inline void jpc_mqdec_setcurctx(jpc_mqdec_t *dec, unsigned ctxno)
{
         dec->curctx = &dec->ctxs[ctxno];
}

/* Set the state information for all contexts of a MQ decoder. */
void jpc_mqdec_setctxs(const jpc_mqdec_t *dec, unsigned numctxs, const jpc_mqctx_t *ctxs);

/******************************************************************************\
* Functions/macros for decoding bits.
\******************************************************************************/

/* Decode a symbol. */
#ifdef NDEBUG
```

```
#define jpc_mqdec_getbit(dec) \
        jpc_mqdec_getbit_macro(dec)
#else
#define jpc_mqdec_getbit(dec) \
        jpc_mqdec_getbit_func(dec)
#endif

/* Decode a symbol (assuming an unskewed probability distribution). */
#ifdef NDEBUG
#define jpc_mqdec_getbitnoskew(dec) \
        jpc_mqdec_getbit_macro(dec)
#else
#define jpc_mqdec_getbitnoskew(dec) \
        jpc_mqdec_getbit_func(dec)
#endif

/******************************************************************************\
* Functions/macros for debugging.
\******************************************************************************/

/* Dump the MQ decoder state for debugging. */
void jpc_mqdec_dump(const jpc_mqdec_t *dec);

/******************************************************************************\
* EVERYTHING BELOW THIS POINT IS IMPLEMENTATION SPECIFIC AND NOT PART OF THE
* APPLICATION INTERFACE.  DO NOT RELY ON ANY OF THE INTERNAL FUNCTIONS/MACROS
* GIVEN BELOW.
\******************************************************************************/

bool jpc_mqdec_mpsexchrenormd(jpc_mqdec_t *dec);
bool jpc_mqdec_lpsexchrenormd(jpc_mqdec_t *dec);

JAS_FORCE_INLINE
static bool jpc_mqdec_getbit_macro(jpc_mqdec_t *dec)
{
        const jpc_mqstate_t *const state = *dec->curctx;

        dec->areg -= state->qeval;

        if (dec->creg >= (uint_least32_t)state->qeval << 16) {
                dec->creg -= (uint_least32_t)state->qeval << 16;
                return dec->areg & 0x8000
                        ? state->mps
                        : jpc_mqdec_mpsexchrenormd(dec);
        } else {
                return jpc_mqdec_lpsexchrenormd(dec);
        }
}

JAS_FORCE_INLINE
static bool jpc_mqdec_mpsexchange(uint_least32_t areg, uint_least32_t delta, const jpc_mqstate_t **curctx)
{
        if (areg < delta) {
                const jpc_mqstate_t *state = *curctx;
                /* LPS decoded. */
                *curctx = state->nlps;
                return !state->mps;
        } else {
                const jpc_mqstate_t *state = *curctx;
                /* MPS decoded. */
                *curctx = state->nmps;
                return state->mps;
        }
}

JAS_FORCE_INLINE
static bool jpc_mqdec_lpsexchange(uint_least32_t *areg_p, uint_least32_t delta, const jpc_mqstate_t
    **curctx)
{
        if (*areg_p >= delta) {
                const jpc_mqstate_t *state = *curctx;
                *areg_p = delta;
                *curctx = state->nlps;
                return !state->mps;
        } else {
                const jpc_mqstate_t *state = *curctx;
                *areg_p = delta;
                *curctx = state->nmps;
                return state->mps;
        }
}
```

```

bool jpc_mqdec_getbit_func(jpc_mqdec_t *dec);

#endif
```

16.58 jpc_mqenc.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * MQ Arithmetic Encoder
 *
 * $Id$
 */

#ifndef JPC_MQENC_H
```

```
#define JPC_MQENC_H

/*****************************************************************************\
* Includes.
\*****************************************************************************/

#include "jasper/jas_types.h"
#include "jasper/jas_stream.h"

#include "jpc_mqcod.h"

#include <stdio.h>

/*****************************************************************************\
* Constants.
\*****************************************************************************/

/*
 * Termination modes.
 */

#define JPC_MQENC_DEFTERM       0       /* default termination */
#define JPC_MQENC_PTERM         1       /* predictable termination */

/*****************************************************************************\
* Types.
\*****************************************************************************/

/* MQ arithmetic encoder class. */

typedef struct {

        /* The C register. */
        uint_least32_t creg;

        /* The A register. */
        uint_least32_t areg;

        /* The CT register. */
        uint_least32_t ctreg;

        /* The maximum number of contexts. */
        unsigned maxctxs;

        /* The per-context information. */
        const jpc_mqstate_t **ctxs;

        /* The current context. */
        const jpc_mqstate_t **curctx;

        /* The stream for encoder output. */
        jas_stream_t *out;

        /* The byte buffer (i.e., the B variable in the standard). */
        int_least16_t outbuf;

        /* The last byte output. */
        int_least16_t lastbyte;

        /* The error indicator. */
        bool err;

} jpc_mqenc_t;

/* MQ arithmetic encoder state information. */

typedef struct {

        /* The A register. */
        unsigned areg;

        /* The C register. */
        unsigned creg;

        /* The CT register. */
        unsigned ctreg;

        /* The last byte output by the encoder. */
        int lastbyte;

} jpc_mqencstate_t;
```

```

/*****************************************************************************\
* Functions/macros for construction and destruction.
\*****************************************************************************/

/* Create a MQ encoder. */
jpc_mqenc_t *jpc_mqenc_create(unsigned maxctxs, jas_stream_t *out);

/* Destroy a MQ encoder. */
void jpc_mqenc_destroy(jpc_mqenc_t *enc);

/*****************************************************************************\
* Functions/macros for initialization.
\*****************************************************************************/

/* Initialize a MQ encoder. */
void jpc_mqenc_init(jpc_mqenc_t *enc);

/*****************************************************************************\
* Functions/macros for context manipulation.
\*****************************************************************************/

/* Set the current context. */
static inline void jpc_mqenc_setcurctx(jpc_mqenc_t *enc, unsigned ctxno) {
        enc->curctx = &enc->ctxs[ctxno];
}

/* Set the state information for multiple contexts. */
void jpc_mqenc_setctxs(jpc_mqenc_t *enc, unsigned numctxs, const jpc_mqctx_t *ctxs);

/*****************************************************************************\
* Miscellaneous functions/macros.
\*****************************************************************************/

/* Get the error state of a MQ encoder. */
static inline bool jpc_mqenc_error(const jpc_mqenc_t *enc) {
        return enc->err;
}

/* Get the current encoder state. */
void jpc_mqenc_getstate(const jpc_mqenc_t *enc, jpc_mqencstate_t *state);

/* Terminate the code. */
int jpc_mqenc_flush(jpc_mqenc_t *enc, int termmode);

/*****************************************************************************\
* Functions/macros for encoding bits.
\*****************************************************************************/

/*****************************************************************************\
* Functions/macros for debugging.
\*****************************************************************************/

int jpc_mqenc_dump(const jpc_mqenc_t *mqenc);

/*****************************************************************************\
* Implementation-specific details.
\*****************************************************************************/

/* Note: These function prototypes are included only to satisfy the
  needs of the mqenc_putbit_macro macro.  Do not call any of these
  functions directly. */
int jpc_mqenc_codemps2(jpc_mqenc_t *enc);
int jpc_mqenc_codelps(jpc_mqenc_t *enc);

int jpc_mqenc_putbit(jpc_mqenc_t *enc, bool bit);

#endif
```

16.59 jpc_qmfb.h

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

```

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * Quadrature Mirror-Image Filter Bank (QMFB) Routines
 *
 * $Id$
 */

#ifndef JPC_QMFB_H
#define JPC_QMFB_H

/******************************************************************************\
* Includes.
\******************************************************************************/

#include "jpc_fix.h"

/******************************************************************************\
* Constants.
\******************************************************************************/

/* QMFB IDs. */
#define JPC_QMFB1D_FT   1       /* 5/3 */
#define JPC_QMFB1D_NS   2       /* 9/7 */

/******************************************************************************\
```

```
* Types.
\*****************************************************************************/

/*****************************************************************************\
* Functions.
\*****************************************************************************/

#if !defined(JPC_QMFB_COLGRPSIZE)
/* The number of columns to group together during the vertical processing
stage of the wavelet transform. */
/* The default value for this parameter is probably not optimal for
any particular platform.  Hopefully, it is not too unreasonable, however. */
#define JPC_QMFB_COLGRPSIZE     16
#endif

typedef struct {
        int (*analyze)(jpc_fix_t *, int, int, int, int, int);
        int (*synthesize)(jpc_fix_t *, int, int, int, int, int);
        const double *lpenergywts;
        const double *hpenergywts;
} jpc_qmfb2d_t;

extern const jpc_qmfb2d_t jpc_ft_qmfb2d;
extern const jpc_qmfb2d_t jpc_ns_qmfb2d;

#endif
```

16.60 jpc_t1cod.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * $Id$
 */

#ifndef JPC_T1COD_H
#define JPC_T1COD_H

/******************************************************************************\
* Includes.
\******************************************************************************/

#include "jpc_fix.h"
#include "jpc_mqcod.h"
#include "jpc_tsfb.h"
#include "jasper/jas_math.h"

/******************************************************************************\
* Constants.
\******************************************************************************/

/* The number of bits used to index into various lookup tables. */
#define JPC_NMSEDEC_BITS        7
#define JPC_NMSEDEC_FRACBITS    (JPC_NMSEDEC_BITS - 1)

/*
 * Segment types.
 */

enum jpc_segtype {
        JPC_SEG_INVALID,

        /* MQ. */
        JPC_SEG_MQ,

        /* Raw. */
         JPC_SEG_RAW,
};

/* The nominal word size. */
#define JPC_PREC        32

/* Tier-1 coding pass types. */
enum jpc_passtype {
         JPC_SIGPASS, /*< significance */
         JPC_REFPASS, /*< refinement */
         JPC_CLNPASS, /*< cleanup */
};

/*
 * Per-sample state information for tier-1 coding.
 */

/* The northeast neighbour has been found to be significant. */
#define JPC_NESIG       0x0001
/* The southeast neighbour has been found to be significant. */
#define JPC_SESIG       0x0002
/* The southwest neighbour has been found to be significant. */
#define JPC_SWSIG       0x0004
/* The northwest neighbour has been found to be significant. */
#define JPC_NWSIG       0x0008
/* The north neighbour has been found to be significant. */
#define JPC_NSIG        0x0010
/* The east neighbour has been found to be significant. */
#define JPC_ESIG        0x0020
/* The south neighbour has been found to be significant. */
#define JPC_SSIG        0x0040
```

```
/* The west neighbour has been found to be significant. */
#define JPC_WSIG        0x0080
/* The significance mask for 8-connected neighbours. */
#define JPC_OTHSIGMSK \
        (JPC_NSIG | JPC_NESIG | JPC_ESIG | JPC_SESIG | JPC_SSIG | JPC_SWSIG | JPC_WSIG | JPC_NWSIG)
/* The significance mask for 4-connected neighbours. */
#define JPC_PRIMSIGMSK  (JPC_NSIG | JPC_ESIG | JPC_SSIG | JPC_WSIG)

/* The north neighbour is negative in value. */
#define JPC_NSGN        0x0100
/* The east neighbour is negative in value. */
#define JPC_ESGN        0x0200
/* The south neighbour is negative in value. */
#define JPC_SSGN        0x0400
/* The west neighbour is negative in value. */
#define JPC_WSGN        0x0800
/* The sign mask for 4-connected neighbours. */
#define JPC_SGNMSK      (JPC_NSGN | JPC_ESGN | JPC_SSGN | JPC_WSGN)

/* This sample has been found to be significant. */
#define JPC_SIG         0x1000
/* The sample has been refined. */
#define JPC_REFINE      0x2000
/* This sample has been processed during the significance pass. */
#define JPC_VISIT       0x4000

/* The number of aggregation contexts. */
#define JPC_NUMAGGCTXS  1
/* The number of zero coding contexts. */
#define JPC_NUMZCCTXS   9
/* The number of magnitude contexts. */
#define JPC_NUMMAGCTXS  3
/* The number of sign coding contexts. */
#define JPC_NUMSCCTXS   5
/* The number of uniform contexts. */
#define JPC_NUMUCTXS    1

/* The context ID for the first aggregation context. */
#define JPC_AGGCTXNO    0
/* The context ID for the first zero coding context. */
#define JPC_ZCCTXNO             (JPC_AGGCTXNO + JPC_NUMAGGCTXS)
/* The context ID for the first magnitude context. */
#define JPC_MAGCTXNO    (JPC_ZCCTXNO + JPC_NUMZCCTXS)
/* The context ID for the first sign coding context. */
#define JPC_SCCTXNO             (JPC_MAGCTXNO + JPC_NUMMAGCTXS)
/* The context ID for the first uniform context. */
#define JPC_UCTXNO              (JPC_SCCTXNO + JPC_NUMSCCTXS)
/* The total number of contexts. */
#define JPC_NUMCTXS             (JPC_UCTXNO + JPC_NUMUCTXS)

/******************************************************************************\
* External data.
\******************************************************************************/

/* These lookup tables are used by various macros/functions. */
/* Do not access these lookup tables directly. */
extern uint_least8_t jpc_zcctxnolut[];
extern bool jpc_spblut[];
extern uint_least8_t jpc_scctxnolut[];
extern uint_least8_t jpc_magctxnolut[];
extern jpc_fix_t jpc_refnmsedec[];
extern jpc_fix_t jpc_signmsedec[];
extern jpc_fix_t jpc_refnmsedec0[];
extern jpc_fix_t jpc_signmsedec0[];

/* The initial settings for the MQ contexts. */
extern jpc_mqctx_t jpc_mqctxs[];

/******************************************************************************\
* Functions and macros.
\******************************************************************************/

/* Arithmetic shift right (with ability to shift left also). */
JAS_ATTRIBUTE_CONST
static inline jpc_fix_t JPC_ASR(jpc_fix_t x, int n)
{
        return n >= 0
                ? x » n
                : x « -n;
}

```

```
/* Get the zero coding context. */
JAS_ATTRIBUTE_CONST
static inline uint_least8_t JPC_GETZCCTXNO(unsigned f, enum jpc_tsfb_orient orient)
{
        return jpc_zcctxnolut[((unsigned)orient « 8) | (f & JPC_OTHSIGMSK)];
}

/* Get the sign prediction bit. */
JAS_ATTRIBUTE_CONST
static inline bool JPC_GETSPB(unsigned f)
{
        return jpc_spblut[(f & (JPC_PRIMSIGMSK | JPC_SGNMSK)) » 4];
}

/* Get the sign coding context. */
JAS_ATTRIBUTE_CONST
static inline uint_least8_t JPC_GETSCCTXNO(unsigned f)
{
        return jpc_scctxnolut[(f & (JPC_PRIMSIGMSK | JPC_SGNMSK)) » 4];
}

/* Get the magnitude context. */
JAS_ATTRIBUTE_CONST
static inline uint_least8_t JPC_GETMAGCTXNO(unsigned f)
{
        return jpc_magctxnolut[(f & JPC_OTHSIGMSK) | (((f & JPC_REFINE) != 0) « 11)];
}

/* Get the normalized MSE reduction for significance passes. */
JAS_ATTRIBUTE_CONST
static inline jpc_fix_t JPC_GETSIGNMSEDEC(jpc_fix_t x, int bitpos)
{
        return bitpos > JPC_NMSEDEC_FRACBITS
                ? jpc_signmsedec[JPC_ASR(x, bitpos - JPC_NMSEDEC_FRACBITS) & JAS_ONES(JPC_NMSEDEC_BITS)]
                : jpc_signmsedec0[JPC_ASR(x, bitpos - JPC_NMSEDEC_FRACBITS) & JAS_ONES(JPC_NMSEDEC_BITS)];
}

/* Get the normalized MSE reduction for refinement passes. */
JAS_ATTRIBUTE_CONST
static inline jpc_fix_t JPC_GETREFNMSEDEC(jpc_fix_t x, int bitpos)
{
        return bitpos > JPC_NMSEDEC_FRACBITS
                ? jpc_refnmsedec[JPC_ASR(x, bitpos - JPC_NMSEDEC_FRACBITS) & JAS_ONES(JPC_NMSEDEC_BITS)]
                : jpc_refnmsedec0[JPC_ASR(x, bitpos - JPC_NMSEDEC_FRACBITS) & JAS_ONES(JPC_NMSEDEC_BITS)];
}

/* Update the per-sample state information. */
static inline void JPC_UPDATEFLAGS4(jpc_fix_t *fp, unsigned rowstep, bool s, bool vcausalflag)
{
        jpc_fix_t *np = fp - rowstep;
        jpc_fix_t *sp = fp + rowstep;
        if (vcausalflag) {
                sp[-1] |= JPC_NESIG;
                sp[1] |= JPC_NWSIG;
                if (s) {
                        *sp |= JPC_NSIG | JPC_NSGN;
                        fp[-1] |= JPC_ESIG | JPC_ESGN;
                        fp[1] |= JPC_WSIG | JPC_WSGN;
                } else {
                        *sp |= JPC_NSIG;
                        fp[-1] |= JPC_ESIG;
                        fp[1] |= JPC_WSIG;
                }
        } else {
                np[-1] |= JPC_SESIG;
                np[1] |= JPC_SWSIG;
                sp[-1] |= JPC_NESIG;
                sp[1] |= JPC_NWSIG;
                if (s) {
                        *np |= JPC_SSIG | JPC_SSGN;
                        *sp |= JPC_NSIG | JPC_NSGN;
                        fp[-1] |= JPC_ESIG | JPC_ESGN;
                        fp[1] |= JPC_WSIG | JPC_WSGN;
                } else {
                        *np |= JPC_SSIG;
                        *sp |= JPC_NSIG;
                        fp[-1] |= JPC_ESIG;
                        fp[1] |= JPC_WSIG;
                }
        }
}
```

```

/* Initialize the lookup tables used by the codec. */
void jpc_initluts(void);

void jpc_initmqctxs(void);

/* Get the nominal gain associated with a particular band. */
JAS_ATTRIBUTE_CONST
unsigned JPC_NOMINALGAIN(unsigned qmfbid, unsigned numlvls, unsigned lvlno, enum jpc_tsfb_orient orient);

/* Get the coding pass type. */
JAS_ATTRIBUTE_CONST
enum jpc_passtype JPC_PASSTYPE(unsigned passno);

/* Get the segment type. */
JAS_ATTRIBUTE_CONST
enum jpc_segtype JPC_SEGTYPE(unsigned passno, unsigned firstpassno, bool bypass);

/* Get the number of coding passess in the segment. */
JAS_ATTRIBUTE_CONST
unsigned JPC_SEGPASSCNT(unsigned passno, unsigned firstpassno, unsigned numpasses, bool bypass,
  bool termall);

/* Is the coding pass terminated? */
JAS_ATTRIBUTE_CONST
bool JPC_ISTERMINATED(unsigned passno, unsigned firstpassno, unsigned numpasses, bool termall,
  bool lazy);

#endif
```

16.61 jpc_t1dec.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * Tier 1 Decoder
 *
 * $Id$
 */

#ifndef JPC_T1DEC_H
#define JPC_T1DEC_H

/******************************************************************************\
* Includes.
\******************************************************************************/

#include "jpc_dec.h"

/******************************************************************************\
* Functions.
\******************************************************************************/

/* Decode all of the code blocks for a particular tile. */
int jpc_dec_decodecblks(jpc_dec_t *dec, jpc_dec_tile_t *tile);

#endif
```

16.62 jpc_t1enc.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * Tier 1 Encoder
 *
 * $Id$
 */

#ifndef JPC_T1ENC_H
#define JPC_T1ENC_H

/******************************************************************************\
* Includes.
\******************************************************************************/

#include "jpc_enc.h"

/******************************************************************************\
* Functions.
\******************************************************************************/

/* Encode all of the code blocks. */
int jpc_enc_enccblks(jpc_enc_t *enc);

#endif
```

16.63 jpc_t2cod.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * Tier-2 Coding Library
 *
 * $Id$
 */

#ifndef JPC_T2COD_H
#define JPC_T2COD_H

/******************************************************************************\
* Includes.
\******************************************************************************/

#include "jpc_cs.h"

/******************************************************************************\
* Types.
\******************************************************************************/

/* Progression change list. */

typedef struct {

        /* The number of progression changes. */
        unsigned numpchgs;

        /* The maximum number of progression changes that can be accomodated
          without growing the progression change array. */
        unsigned maxpchgs;

        /* The progression changes. */
        jpc_pchg_t **pchgs;

} jpc_pchglist_t;

/* Packet iterator per-resolution-level information. */

typedef struct {

```

```
        /* The number of precincts. */
        unsigned numprcs;

        /* The last layer processed for each precinct. */
        unsigned *prclyrnos;

        /* The precinct width exponent. */
        unsigned prcwidthexpn;

        /* The precinct height exponent. */
        unsigned prcheightexpn;

        /* The number of precincts spanning the resolution level in the horizontal
          direction. */
        unsigned numhprcs;

} jpc_pirlvl_t;

/* Packet iterator per-component information. */

typedef struct {

        /* The number of resolution levels. */
        unsigned numrlvls;

        /* The per-resolution-level information. */
        jpc_pirlvl_t *pirlvls;

        /* The horizontal sampling period. */
        uint_fast32_t hsamp;

        /* The vertical sampling period. */
        uint_fast32_t vsamp;

} jpc_picomp_t;

/* Packet iterator class. */

typedef struct {

        /* The number of layers. */
        unsigned numlyrs;

        /* The number of resolution levels. */
        unsigned maxrlvls;

        /* The number of components. */
        unsigned numcomps;

        /* The per-component information. */
        jpc_picomp_t *picomps;

        /* The current component. */
        jpc_picomp_t *picomp;

        /* The current resolution level. */
        jpc_pirlvl_t *pirlvl;

        /* The number of the current component. */
        unsigned compno;

        /* The number of the current resolution level. */
        unsigned rlvlno;

        /* The number of the current precinct. */
        unsigned prcno;

        /* The number of the current layer. */
        unsigned lyrno;

        /* The x-coordinate of the current position. */
        uint_fast32_t x;

        /* The y-coordinate of the current position. */
        uint_fast32_t y;

        /* The horizontal step size. */
        uint_fast32_t xstep;

        /* The vertical step size. */
        uint_fast32_t ystep;
```

```
        /* The x-coordinate of the top-left corner of the tile on the reference
          grid. */
        uint_fast32_t xstart;

        /* The y-coordinate of the top-left corner of the tile on the reference
          grid. */
        uint_fast32_t ystart;

        /* The x-coordinate of the bottom-right corner of the tile on the
          reference grid (plus one). */
        uint_fast32_t xend;

        /* The y-coordinate of the bottom-right corner of the tile on the
          reference grid (plus one). */
        uint_fast32_t yend;

        /* The current progression change. */
        const jpc_pchg_t *pchg;

        /* The progression change list. */
        jpc_pchglist_t *pchglist;

        /* The progression to use in the absense of explicit specification. */
        jpc_pchg_t defaultpchg;

        /* The current progression change number. */
        int pchgno;

        /* Is this the first time in the current progression volume? */
        bool prgvolfirst;

        /* Is the current iterator value valid? */
        bool valid;

        /* The current packet number. */
        int pktno;

} jpc_pi_t;

/*****************************************************************************\
* Functions/macros for packet iterators.
\*****************************************************************************/

/* Create a packet iterator. */
jpc_pi_t *jpc_pi_create0(void);

/* Destroy a packet iterator. */
void jpc_pi_destroy(jpc_pi_t *pi);

/* Add a progression change to a packet iterator. */
int jpc_pi_addpchg(jpc_pi_t *pi, jpc_pocpchg_t *pchg);

/* Prepare a packet iterator for iteration. */
int jpc_pi_init(jpc_pi_t *pi);

/* Set the iterator to the first packet. */
int jpc_pi_begin(jpc_pi_t *pi);

/* Proceed to the next packet in sequence. */
int jpc_pi_next(jpc_pi_t *pi);

/* Get the index of the current packet. */
#define jpc_pi_getind(pi)       ((pi)->pktno)

/* Get the component number of the current packet. */
#define jpc_pi_cmptno(pi)       (assert(pi->valid), (pi)->compno)

/* Get the resolution level of the current packet. */
#define jpc_pi_rlvlno(pi)       (assert(pi->valid), (pi)->rlvlno)

/* Get the layer number of the current packet. */
#define jpc_pi_lyrno(pi)        (assert(pi->valid), (pi)->lyrno)

/* Get the precinct number of the current packet. */
#define jpc_pi_prcno(pi)        (assert(pi->valid), (pi)->prcno)

/* Get the progression order for the current packet. */
#define jpc_pi_prg(pi)  (assert(pi->valid), (pi)->pchg->prgord)

/*****************************************************************************\
```

```
* Functions/macros for progression change lists.
\*****************************************************************************/

/* Create a progression change list. */
jpc_pchglist_t *jpc_pchglist_create(void);

/* Destroy a progression change list. */
void jpc_pchglist_destroy(jpc_pchglist_t *pchglist);

/* Insert a new element into a progression change list. */
int jpc_pchglist_insert(jpc_pchglist_t *pchglist, int pchgno, jpc_pchg_t *pchg);

/* Remove an element from a progression change list. */
jpc_pchg_t *jpc_pchglist_remove(jpc_pchglist_t *pchglist, unsigned pchgno);

/* Get an element from a progression change list. */
JAS_ATTRIBUTE_PURE
const jpc_pchg_t *jpc_pchglist_get(const jpc_pchglist_t *pchglist, unsigned pchgno);

/* Copy a progression change list. */
jpc_pchglist_t *jpc_pchglist_copy(const jpc_pchglist_t *pchglist);

/* Get the number of elements in a progression change list. */
JAS_ATTRIBUTE_PURE
unsigned jpc_pchglist_numpchgs(const jpc_pchglist_t *pchglist);

/*****************************************************************************\
* Functions/macros for progression changes.
\*****************************************************************************/

/* Destroy a progression change. */
void jpc_pchg_destroy(jpc_pchg_t *pchg);

/* Copy a progression change. */
jpc_pchg_t *jpc_pchg_copy(const jpc_pchg_t *pchg);

#endif
```

16.64 jpc_t2dec.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * Tier 2 Decoder
 *
 * $Id$
 */

#ifndef JPC_T2DEC_H
#define JPC_T2DEC_H

/*****************************************************************************\
* Includes.
\*****************************************************************************/

#include "jasper/jas_stream.h"

#include "jpc_dec.h"
#include "jpc_t2cod.h"

/*****************************************************************************\
* Functions.
\*****************************************************************************/

/* Decode the packets for a tile-part. */
int jpc_dec_decodepkts(jpc_dec_t *dec, jas_stream_t *pkthdrstream,
  jas_stream_t *in);

/* Create a packet iterator for the decoder. */
jpc_pi_t *jpc_dec_pi_create(jpc_dec_t *dec, jpc_dec_tile_t *tile);

#endif
```

16.65 jpc_t2enc.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * Tier 2 Encoder
 *
 * $Id$
 */

#ifndef JPC_T2ENC_H
#define JPC_T2ENC_H

/******************************************************************************\
* Includes.
\******************************************************************************/

#include "jasper/jas_stream.h"

#include "jpc_enc.h"
#include "jpc_t2cod.h"

/******************************************************************************\
* Functions.
\******************************************************************************/

/* Encode the packets for a tile. */
int jpc_enc_encpkts(jpc_enc_t *enc, jas_stream_t *out);

/* Encode the specified packet. */
int jpc_enc_encpkt(jpc_enc_t *enc, jas_stream_t *out, unsigned compno,
  unsigned lvlno, unsigned prcno, unsigned lyrno);

/* Save the tier-2 coding state. */
void jpc_save_t2state(jpc_enc_t *enc);

/* Restore the tier-2 coding state. */
void jpc_restore_t2state(jpc_enc_t *enc);

/* Initialize the tier-2 coding state. */
```

```
void jpc_init_t2state(jpc_enc_t *enc, bool raflag);

/* Create a packet iterator for the encoder. */
jpc_pi_t *jpc_enc_pi_create(jpc_enc_cp_t *cp, jpc_enc_tile_t *tile);

#endif
```

16.66 jpc_tagtree.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * Tag Tree Library
 *
 * $Id$
 */
```

```
#ifndef JPC_TAGTREE_H
#define JPC_TAGTREE_H

/******************************************************************************\
* Includes
\******************************************************************************/

#include <stdio.h>

#include "jpc_bs.h"

/******************************************************************************\
* Constants
\******************************************************************************/

/* The maximum allowable depth for a tag tree. */
#define JPC_TAGTREE_MAXDEPTH    32

/******************************************************************************\
* Types
\******************************************************************************/

/*
 * Tag tree node.
 */

typedef struct jpc_tagtreenode_ {

        /* The parent of this node. */
        struct jpc_tagtreenode_ *parent_;

        /* The value associated with this node. */
        int value_;

        /* The lower bound on the value associated with this node. */
        int low_;

        /* A flag indicating if the value is known exactly. */
        int known_;

} jpc_tagtreenode_t;

/*
 * Tag tree.
 */

typedef struct {

        /* The number of leaves in the horizontal direction. */
        int numleafsh_;

        /* The number of leaves in the vertical direction. */
        int numleafsv_;

        /* The total number of nodes in the tree. */
        int numnodes_;

        /* The nodes. */
        jpc_tagtreenode_t *nodes_;

} jpc_tagtree_t;

/******************************************************************************\
* Functions.
\******************************************************************************/

/* Create a tag tree. */
jpc_tagtree_t *jpc_tagtree_create(int numleafsh, int numleafsv);

/* Destroy a tag tree. */
void jpc_tagtree_destroy(jpc_tagtree_t *tree);

/* Copy data from one tag tree to another. */
void jpc_tagtree_copy(jpc_tagtree_t *dsttree, const jpc_tagtree_t *srctree);

/* Reset the tag tree state. */
void jpc_tagtree_reset(jpc_tagtree_t *tree);

/* Set the value associated with a particular leaf node of a tag tree. */
void jpc_tagtree_setvalue(jpc_tagtree_t *tree, jpc_tagtreenode_t *leaf,
```

```
  int value);

/* Get a pointer to a particular leaf node. */
JAS_ATTRIBUTE_PURE
jpc_tagtreenode_t *jpc_tagtree_getleaf(jpc_tagtree_t *tree, int n);

/* Invoke the tag tree decoding procedure. */
int jpc_tagtree_decode(jpc_tagtree_t *tree, jpc_tagtreenode_t *leaf,
  int threshold, jpc_bitstream_t *in);

/* Invoke the tag tree encoding procedure. */
int jpc_tagtree_encode(jpc_tagtree_t *tree, jpc_tagtreenode_t *leaf,
  int threshold, jpc_bitstream_t *out);

/* Dump a tag tree (for debugging purposes). */
void jpc_tagtree_dump(const jpc_tagtree_t *tree, FILE *out);

#endif
```

16.67 jpc_tsfb.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * Tree-Structured Filter Bank (TSFB) Library
 *
 * $Id$
 */

#ifndef JPC_TSFB_H
#define JPC_TSFB_H

/******************************************************************************\
* Includes.
\******************************************************************************/

#include "jasper/jas_seq.h"
#include "jasper/jas_types.h"

#include "jpc_fix.h"
#include "jpc_qmfb.h"

/******************************************************************************\
* Constants.
\******************************************************************************/

#define JPC_TSFB_MAXBANDS       (JPC_TSFB_MAXDEPTH * 3 + 1)
#define JPC_TSFB_MAXDEPTH       32
#define JPC_TSFB_RITIMODE       JPC_QMFB1D_RITIMODE

enum jpc_tsfb_orient {
        JPC_TSFB_LL = 0,
        JPC_TSFB_LH = 1,
        JPC_TSFB_HL = 2,
        JPC_TSFB_HH = 3,
};

/******************************************************************************\
* Types.
\******************************************************************************/

typedef struct {
        int xstart;
        int ystart;
        int xend;
        int yend;
        enum jpc_tsfb_orient orient;
        int locxstart;
        int locystart;
        int locxend;
        int locyend;
        jpc_fix_t synenergywt;
} jpc_tsfb_band_t;

typedef struct {
        unsigned numlvls;
        const jpc_qmfb2d_t *qmfb;
} jpc_tsfb_t;

/******************************************************************************\
* Functions.
\******************************************************************************/

/* Create a TSFB. */
jpc_tsfb_t *jpc_cod_gettsfb(unsigned qmfbid, unsigned numlevels);

/* Destroy a TSFB. */
void jpc_tsfb_destroy(jpc_tsfb_t *tsfb);

/* Perform analysis. */
int jpc_tsfb_analyze(jpc_tsfb_t *tsfb, jas_seq2d_t *x);

/* Perform synthesis. */
int jpc_tsfb_synthesize(jpc_tsfb_t *tsfb, jas_seq2d_t *x);

/* Get band information for a TSFB. */
int jpc_tsfb_getbands(jpc_tsfb_t *tsfb, uint_fast32_t xstart,
```

```
  uint_fast32_t ystart, uint_fast32_t xend, uint_fast32_t yend,
  jpc_tsfb_band_t *bands);

#endif
```

16.68 jpc_util.h

```
/*
 * Copyright (c) 2001-2002 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JPC_UTIL_H
#define JPC_UTIL_H

#include "jpc_fix.h"

#include "jasper/jas_seq.h"

/* Parse a comma separated list of real numbers into an array of doubles. */
int jpc_atoaf(const char *s, int *numvalues, double **values);
```

```

/* Upsample a sequence. */
jas_seq_t *jpc_seq_upsample(const jas_seq_t *seq, int n);

/* Convolve two sequences. */
jas_seq_t *jpc_seq_conv(const jas_seq_t *seq0, const jas_seq_t *seq1);

/* Compute the norm of a sequence. */
JAS_ATTRIBUTE_PURE
jpc_fix_t jpc_seq_norm(const jas_seq_t *x);

#endif
```

16.69 jpg_cod.h

```
/*
 * Copyright (c) 2001-2002 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
```

```
 * JPG Format Library
 *
 * $Id$
 */

#ifndef JPG_COD_H
#define JPG_COD_H

/*****************************************************************************\
* Includes.
\*****************************************************************************/

/*****************************************************************************\
* Constants.
\*****************************************************************************/

#define JPG_MAGIC       0xffd8
#define JPG_MAGICLEN 2

#endif
```

16.70 jpg_enc.h

```
/*
 * Copyright (c) 2002 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JPG_ENC_H
#define JPG_ENC_H

typedef struct {
        int numcmpts;
        int cmpts[4];
} jpg_enc_t;

#endif
```

16.71 jpg_jpeglib.h

```
/*
 * Copyright (c) 2001-2002 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
```

```
 * __END_OF_JASPER_LICENSE__
 */

#ifndef JPG_JPEGLIB_H
#define JPG_JPEGLIB_H

/*****************************************************************************\
* Includes.
\*****************************************************************************/

#include <stdio.h>
#include "jasper/jas_types.h"

/* Note: The jpeglib.h header file does not include definitions of
  FILE, size_t, etc. */
#include <jpeglib.h> /* IWYU pragma: export */

#endif
```

16.72 mif_cod.h

```
/*
 * Copyright (c) 2001-2002 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef MIF_COD_H
#define MIF_COD_H

/*****************************************************************************\
* Includes.
\*****************************************************************************/

#include "jasper/jas_types.h"

/*****************************************************************************\
* Constants.
\*****************************************************************************/

#define MIF_MAGIC               0x4d49460a
/* signature */

#define MIF_MAGICLEN    4
/* length of signature in bytes */

/*****************************************************************************\
* Types.
\*****************************************************************************/

/* Per-component information. */

typedef struct {

        int_fast32_t tlx;

        int_fast32_t tly;

        int_fast32_t width;

        int_fast32_t height;

        int_fast32_t sampperx;

        int_fast32_t samppery;

         int_fast16_t prec;

         int_fast16_t sgnd;

         char *data;

} mif_cmpt_t;

/* MIF header. */

typedef struct {

         uint_fast32_t magic;

         int numcmpts;

         int maxcmpts;

         mif_cmpt_t **cmpts;

} mif_hdr_t;

#endif
```

16.73 pgx_cod.h

```
/*
 * Copyright (c) 2001-2002 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
```

```
* JasPer License Version 2.0
*
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * PGX Format Library
 *
 * $Id$
 */

#ifndef PGX_COD_H
#define PGX_COD_H

/******************************************************************************\
* Includes.
\******************************************************************************/

#include <stdio.h>

#include "jasper/jas_types.h"

/******************************************************************************\
* Constants.
\******************************************************************************/

#define PGX_MAGIC       0x5047
#define PGX_MAGICLEN    2

/******************************************************************************\
* Types.
\******************************************************************************/
```

```

typedef struct {

        uint_fast16_t magic;
        /* The signature. */

        bool bigendian;
        /* The byte ordering used. */

        bool sgnd;
        /* The signedness of the samples. */

         uint_fast32_t prec;
         /* The precision of the samples. */

         uint_fast32_t width;
         /* The width of the component. */

         uint_fast32_t height;
         /* The height of the component. */

} pgx_hdr_t;

/*****************************************************************************\
* Functions.
\*****************************************************************************/

void pgx_dumphdr(FILE *out, pgx_hdr_t *hdr);

#endif
```

16.74 pgx_enc.h

```
/*
 * Copyright (c) 2002 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef PGX_ENC_H
#define PGX_ENC_H

typedef struct {
        int cmpt;
} pgx_enc_t;

#endif
```

16.75 pnm_cod.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * Portable Pixmap/Graymap Format Support
 *
 * $Id$
 */

#ifndef PNM_COD_H
#define PNM_COD_H

/*****************************************************************************\
* Includes.
\*****************************************************************************/

#include "jasper/jas_types.h"

/*****************************************************************************\
* Constants.
\*****************************************************************************/

/* Magic numbers. */
#define PNM_MAGIC_TXTPBM        0x5031 /* Text Portable BitMap (P1) */
#define PNM_MAGIC_TXTPGM        0x5032 /* Text Portable GrayMap (P2) */
#define PNM_MAGIC_TXTPPM        0x5033 /* Text Portable PixMap (P3) */
#define PNM_MAGIC_BINPBM        0x5034 /* Binary Portable BitMap (P4) */
#define PNM_MAGIC_BINPGM        0x5035 /* Binary Portable GrayMap (P5) */
#define PNM_MAGIC_BINPPM        0x5036 /* Binary Portable PixMap (P6) */
#define PNM_MAGIC_PAM           0x5037 /* PAM (P7) */

/* Type of image data. */
#define PNM_TYPE_INVALID        0
#define PNM_TYPE_PBM    1 /* BitMap */
#define PNM_TYPE_PGM    2 /* GrayMap */
#define PNM_TYPE_PPM    3 /* PixMap */

/* Format of image data. */
#define PNM_FMT_TXT     0 /* Text */
#define PNM_FMT_BIN     1 /* Binary */

#define PNM_MAXLINELEN  79

#define PNM_TUPLETYPE_UNKNOWN   0
#define PNM_TUPLETYPE_MONO      1
#define PNM_TUPLETYPE_GRAY      2
#define PNM_TUPLETYPE_GRAYA     3
#define PNM_TUPLETYPE_RGB       4
#define PNM_TUPLETYPE_RGBA      5

/*****************************************************************************\
* Types.
\*****************************************************************************/

/* File header. */

typedef struct {

        int magic;
        /* The magic number. */

        int width;
        /* The image width. */

        int height;
        /* The image height. */

        int numcmpts;

        int maxval;
```

```
        /* The maximum allowable sample value. */

#if 0
        int tupletype;
#endif

        bool sgnd;
        /* The sample data is signed. */

} pnm_hdr_t;

/*****************************************************************************\
* Functions.
\*****************************************************************************/

int pnm_type(uint_fast16_t magic);
/* Determine type (i.e., PGM or PPM) from magic number. */

int pnm_fmt(uint_fast16_t magic);
/* Determine format (i.e., text or binary) from magic number. */

int pnm_maxvaltodepth(uint_fast32_t maxval);
/* Determine depth (i.e., precision) from maximum value. */

#define PNM_ONES(n) \
        (((n) < 32) ? ((1UL << (n)) - 1) : 0xffffffffUL)
#endif
```

16.76 pnm_enc.h

```
/*
 * Copyright (c) 2002 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef PNM_ENC_H
#define PNM_ENC_H

typedef struct {
        int numcmpts;
        int cmpts[4];
} pnm_enc_t;

#endif
```

16.77 ras_cod.h

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

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

/*
 * Sun Rasterfile Library
 *
 * $Id$
 */

/*****************************************************************************\
* Sun Rasterfile
\*****************************************************************************/

#ifndef RAS_COD_H
#define RAS_COD_H

/*****************************************************************************\
* Includes.
\*****************************************************************************/

#include "jasper/jas_types.h"

/*****************************************************************************\
* Constants.
\*****************************************************************************/

#define RAS_MAGIC       0x59a66a95
#define RAS_MAGICLEN    4

#define RAS_TYPE_OLD    0
#define RAS_TYPE_STD    1
#define RAS_TYPE_RLE    2

#define RAS_MT_NONE     0
#define RAS_MT_EQUALRGB 1
#define RAS_MT_RAW      2

/*****************************************************************************\
* Types.
\*****************************************************************************/

/* Sun Rasterfile header. */
typedef struct {

        int_fast32_t magic;
        /* signature */

        int_fast32_t width;
        /* width of image (in pixels) */

        int_fast32_t height;
        /* height of image (in pixels) */

        int_fast32_t depth;
        /* number of bits per pixel */

        int_fast32_t length;
        /* length of image data (in bytes) */

        int_fast32_t type;
        /* format of image data */

        int_fast32_t maptype;
        /* colormap type */

        int_fast32_t maplength;
        /* length of colormap data (in bytes) */

} ras_hdr_t;

#define RAS_CMAP_MAXSIZ 256

```

```
/* Color map. */
typedef struct {

        int len;
        /* The number of entries in the color map. */

        int data[RAS_CMAP_MAXSIZ];
        /* The color map data. */

} ras_cmap_t;

/*****************************************************************************\
* Macros.
\*****************************************************************************/

#define RAS_GETBLUE(x)  (((x) » 16) & 0xff)
#define RAS_GETGREEN(x) (((x) » 8) & 0xff)
#define RAS_GETRED(x)   ((x) & 0xff)

#define RAS_BLUE(x)     (((x) & 0xff) « 16)
#define RAS_GREEN(x)    (((x) & 0xff) « 8)
#define RAS_RED(x)      ((x) & 0xff)

#define RAS_ROWSIZE(hdr) \
        ((((hdr)->width * (hdr)->depth + 15) / 16) * 2)
#define RAS_ISRGB(hdr) \
        ((hdr)->depth == 24 || (hdr)->depth == 32)

#define RAS_ONES(n) \
        (((n) == 32) ? 0xffffffffUL : ((1UL « (n)) - 1))

#endif
```

16.78 ras_enc.h

```
/*
 * Copyright (c) 2002 Michael David Adams.
 * All rights reserved.
 */

/* __START_OF_JASPER_LICENSE__
 *
 * JasPer License Version 2.0
 *
 * Copyright (c) 2001-2006 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.
 *
 * __END_OF_JASPER_LICENSE__
 */

#ifndef RAS_ENC_H
#define RAS_ENC_H

typedef struct {
        int numcmpts;
        int cmpts[4];
} ras_enc_t;

#endif
```