

# Getting Started with The i2k Align and i2k Align Retina Toolkits

Charles V. Stewart  
DualAlign LLC  
Clifton Park, NY 12065  
*chuck@dualalign.com*

July 2, 2009

## Overview

Welcome to use of the `i2k Align` and `i2k Align Retina` software. This document gives users instructions on how to run the software and how to activate it once the user has purchased an activation code. The document assumes the software has been downloaded and installed. All of the functionality of `i2k Align` is included in `i2k Align Retina`, but `i2k Align Retina` has additional functionality for aligning and montaging retinal mages. The description that follows uses the name `i2k Align` generically, and references `i2k Align Retina` as needed.

The software may be run in three different ways:

- From the user interface (UI) distributed with the software (available for free by July 15, 2009).
- Using the command-line interface.
- By compiling and running software that calls the functions of the library.

On-line documentation for the UI is available through the UI's main menus. The current document focuses on the latter two.

## Software Location

On Windows the software is by default installed under `c:\Program Files\i2k Align` If `i2k Align Retina` is used, the software is installed under `c:\Program Files\i2k Align Retina`. Currently, the installation includes two header files, `da_error_codes.h` and `da_i2k_align.h`, the library file `da_i2k_align.lib`, and the executable. This will be expanded soon to include the UI code.

On a Mac, the software is placed in several different locations. The command-line executable `i2k_align_exec` and the UI code for `i2k Align` are in `/Applications/i2kAlign`, the library `libda_i2k_align.dylib` is in `/usr/local/lib`, and the two header files `da_error_codes.h` and `da_i2k_align.h` are in `/usr/local/include`. If `i2k Align Retina` is used, the command-line executable and the UI code for `i2k Align Retina` are in `/Applications/i2kAlignRetina` and the library name is `libda_i2k_align_retina.dylib`, with all other paths and header files being the same.

## Running the Command-Line Executable

The command-line executable contains switches for running the library under a variety of options and for activating the software once the user has purchased an activation code. A list of the options may be obtained by typing

```
da_i2k_align_retina_exec -?
```

(The `-?` may be typed at any place on the command-line even if many other arguments are being provided as well.) More details on these switches can be found in the header file `da_i2k_align.h`. The `exec` and the library do extensive error checking on the form of the input, with as much of the checked pushed down into the library as possible.

Here are some examples using the images provided with the installation. These instructions assume the user is in the folder `i2k_align_examples`, delivered with the installer. If the path to the `exec` is not in the user's path list, then the name `da_i2k_align_exec` in the command-line examples below will need to be expanded to indicate the `exec`'s location. In each of the examples where `da_i2k_align_exec` is used, `da_i2k_align_retina_exec` can be used in its place.

1. Basic montaging

```
da_i2k_align_exec hawaii/hawaii.list
```

Builds a montage of the iPhone images whose names are listed in `hawaii.list`. The montage will be stored in the current working directory.

2. Specifying the output location

```
da_i2k_align_exec hawaii/hawaii.list -dir hawaii/results
```

Builds a montage of the iPhone images whose names are listed in `hawaii.list`. The montage will be in subdirectory called `results` created, if necessary, in the `hawaii` directory.

3. Basic alignment:

```
da_i2k_align_exec hawaii/hawaii.list -tool 1 -layout 2 -dir hawaii
```

Aligns the images listed in `hawaii.list`, outputting the mapped images to separate files in the same folder as `hawaii.list`. The cylindrical layout is used. Remember, the layout must be specified (a) for alignment or (b) for montaging of non-retinal images.

4. Montaging thermal images:

```
da_i2k_align_exec thermal/thermal.list -tool 0 -image_type 2 -layout 5
```

(Note that `-tool 0` was not strictly necessary.) The image type here is “non-photographic” which works for thermal and multimodal image sets. Layout option 5 is for affine transformations, which works well for most thermal images because of their low resolution.

5. Aligning multimodal brain MRI's

```
da_i2k_align_exec brain/brain.list -dir brain/results -tool 1 \  
-image_type 2 -layout 6 -apply_masking -save_xforms
```

(this should all be on one line). Here the homography layout is used, but planar and affine would have worked just as well. DualAlign's convex mask calculation algorithm is applied to restrict attention to the primary regions of the images. The estimated transformations and correspondences are output to two files in the output directory `brain/results`, which is created if it does not already exist.

#### 6. Aligning and cropping images for HDR applications

```
da_i2k_align_exec hdr/hdr.list -tool 1 -layout 1 -crop 2 -dir hdr/results
```

Aligns the images and crops them to their overlap.

#### 7. Aligning retinal red-free images and fluorescein angiograms

```
cd angio  
da_i2k_align_retina_exec angio.list -tool 1 -image_type 0 -crop 3 -target IMG_0038.jpg
```

Here, we have moved down into the `angio` folder prior to running the `exec`, and therefore the specification of the file containing the list of images is simpler. The image type is retinal (the 0) and the software produces a set of aligned images, cropped to fit inside the mapped version of `IMG_0038.jpg`.

#### 8. Creating a retinal montage

```
cd ..  
da_i2k_align_retina_exec retina_montage/F.list -image_type 0
```

## Licensing and Activation

When the software is downloaded and started, it will be in trial mode, which allows the user to run it for up to 5 days and 25 runs. During this time, all functionality is available, but the output images and montages have watermarks on them. At the end of this period (or before if the user wishes), the user must purchase an activation code to continue to run the software. The 16-digit code for either `i2k Align` or `i2k Align Retina` may be purchased at

[http://i2align.com/purchase\\_retina\\_software.php](http://i2align.com/purchase_retina_software.php).

Each purchased activation code is for a single workstation.

Once a user has an activation code, it is easy to activate through the UI. It can also be done through the stand-alone `exec`, as follows:

```
da_i2k_align_exec foo.list -activate xxxx-xxxx-xxxx-xxxx
```

where `foo.list` is a dummy argument (the file need not even exist!) and the `-activate` switch is followed by the actual code.

In rare instances, in particular for users behind a particularly stringent firewall, activation will need to be done through a multi-step semi-manual process:

1. Run the command-line exec

```
da_i2k_align_exec foo.list -machine_id
```

2. Email the machine id that is output, together with the 16-digit activation code output, to [support@i2align.com](mailto:support@i2align.com).
3. When the return email is received, save the attached license file onto the hard disk. Any name is fine, but to make the example here concrete, assume it has been saved to `license.txt`.
4. Run the command-line exec (from the folder where `license.txt` is stored) as

```
da_i2k_align_exec foo.list -activate xxxx-xxxx-xxxx-xxxx -manual_file license.txt
```

## The Library

Users who want to write their own programs that link against the `i2k Align` and `i2k Align Retina` libraries will find a straight-forward, C-level application interface described in `da_i2k_align.h`, distributed with the installer. Particular care will be needed to use the activation functions, since each machine the software is installed on will require its own separate activation code.

The software runs on the Mac (10.4 and later) and on Windows (XP and Vista), and versions are also available on Linux machines.

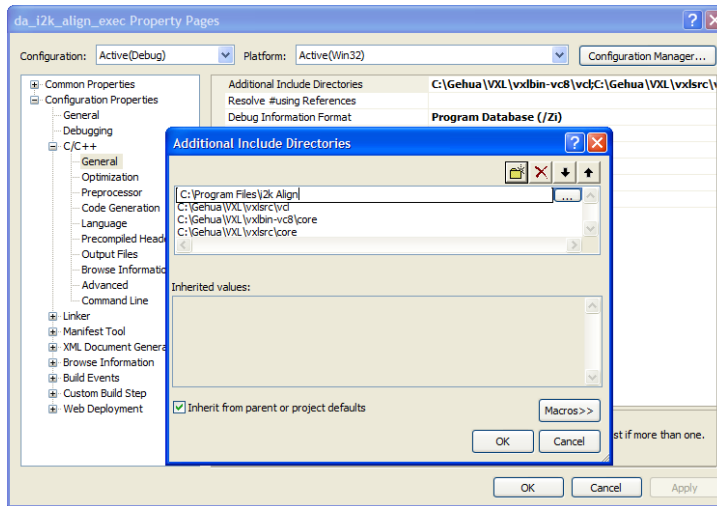
The software on the Mac was compiled using the CMake cross-platform build facility and gcc 4.0.1 to create a universal binary. Users compiling and linking against the library must be sure to locate the two header files in their include path and link the library correctly.

The software for Windows is compiled with Microsoft Visual Studio 2005 SP1 in Release mode. But users may work with other versions of Visual Studio or even other compilers.

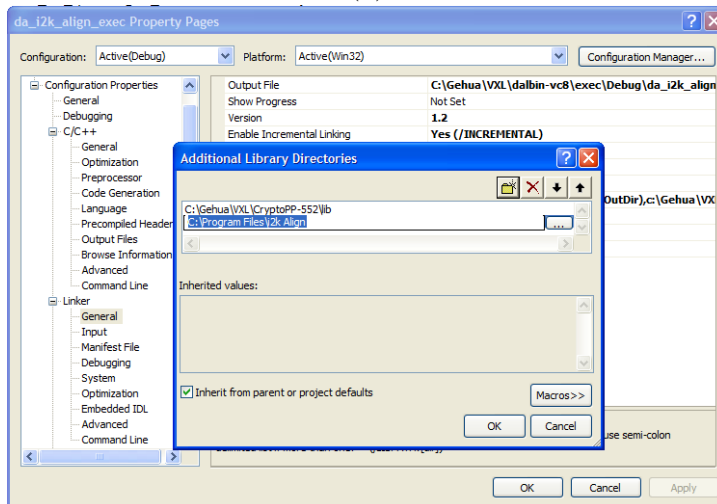
Taking the example of including `i2k Align` into a Visual Studio C++ project, the user must make the following three changes to the project configuration page:

1. Add the installation path to **Additional Include Directories** under **General** of **C/C++** (see (a) in the following figure).
2. Add the installation path to **Additional Library Directories** under **General** of **Linker** (see (b) in the following figure).
3. Add the library name `da_i2k_align.lib` to **Additional Dependencies** under **Input of Linker** (see (b) in the following figure).

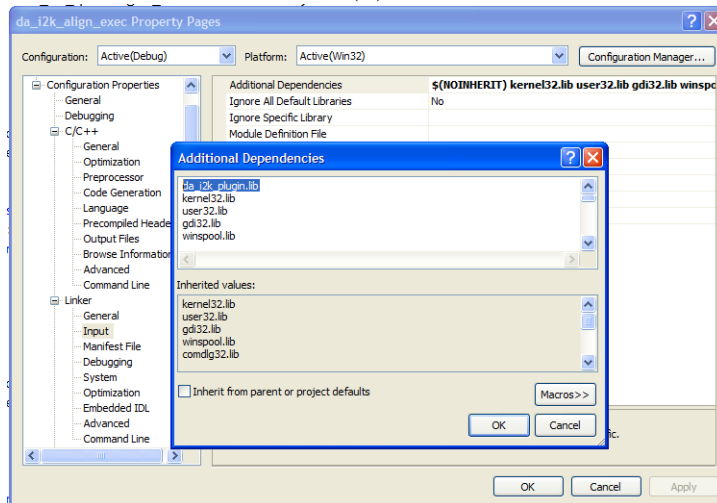
After establishing these settings, the user's code may call any functions in the library. The same idea applies to `i2k Align Retina`, although the library `da_i2k_align_retina.lib` must be used instead. Once project compilation is finished, the user must make sure the `da_i2k_align.dll` file is either in the current working directory or in one of the directories specified in the `PATH` environment variable.



(a)



(b)



(c)