Seeing life in a new light™

CRi MSI File Formats API Version 1.0.0

Application Overview

Introduction

The **CRi MSI File Formats API** is a C interface for extracting data from cri archived files such as cubes (im3), unmixed result sets (umx), spectral libraries (cls), regions of interest (roi), and protocols (pro). The software consists of two components: a software development kit (SDK) used to develop application programs, and a redistributable library DLL file(s) which is used by application programs at run-time. Windows XP SP2 is required to run the software. The dll is compiled with Visual Studio 7.1 and uses the multi-threaded C runtime.

Handles

All the functions in this API take a handle parameter that is a unique identifier created by the software when the user starts the session by opening the archive. The handle should be closed at the end of the session.

Error Codes

All the functions return an error code. A return value other than zero is indication of an error which is documented in the `ErrorCodes.h` file. The function `"GetErrorDescription"` is provided to get the detailed description of the error code.

Memory Management

Some functions need to allocate memory for their output parameters. For example, let us assume the parameter name `"data"` needs memory allocation. There are three possible scenarios regarding this parameter.

1. `"data"` is passed as NULL. The software will ignore the pointer parameter and return other value parameters in the function.
2. `"data"` is pointing to NULL. The software will allocate memory with the `HeapAlloc(GetProcessHeap)` function for the data. The user can deallocate this memory by the `DeallocateMemory` function. Deallocation is done by the `HeapFree(GetProcessHeap)` function.
3. `"data"` is pointing to an allocated buffer. The software will check the size of the allocated buffer and it will return an error code if the size of the buffer is not big enough to hold the requested data.

API Functions

These libraries export C functions so that they can be readily used in a variety of programming environments.

File Functions

`cri_OpenReadArchive`

Open the file and return a handle that can be used to access other archive functions in the api. The full path of the file should be given to the function.

`Cri_CloseArchive`

Close the file.

`cri_GetArchiveType`

Get the type that is stored in the file (e.g., a cube, spectral library, etc.).

Cube Functions

cri_GetCubeData

Get cube data. This is a flexible function that can use a preallocated buffer, or can allocate memory itself. It can be used to get the size of the buffer to allocate. It can also be used to convert pixels from the unit it is stored in to a user-specified unit.

cri_ GetCubeWavelengths

Get all the wavelengths for the cube. If **wavelengths** is null, returns just **numWavelengths**. If **wavelengths** is not null, then **wavelengths** is a preallocated buffer.

cri_GetCubeExposureTimeUS

Get the exposure time in micro seconds. There is an exposure time for each plane in the cube.

cri_GetCubeBinning

Get the camera binning.

cri_GetCubeCameraBitDepth

Get the camera bit depth. Currently, this will be 8 or 12.

cri_GetCubeCameraGainAndOffset

Get the camera gain.

Spectral Library

cri_GetSpectra

Get spectra from spectral library.

Region of Interest

cri_GetMasks

Get all the masks. If **masks** is null, do not retrieve this parameter. The routine will allocate the memory. If **masks** is not null, then this is a preallocated buffer of size **numROIS** and at most **numROIS** will be retrieved. The mask slot in the **cri_Mask** struct is always allocated by this routine.

cri_GetMasksAsLabelImage

Get masks as label image. If **labelImage** is null, do not retrieve the parameter. If **labelImage** is not null, the image is preallocated with the dims passed in through **width** and **height**. A pixel in a label image shows what mask that pixel belongs to. Label **zero** is the background.

Result Set

`cri_GetComponentImages`

Get component images. If **`component`** is null, do not retrieve the parameter. The routine will allocate the memory. If **`pixelUnit!=cri_pixel_unit_unknown`**, then try to convert to the requested pixel unit. The actual pixel unit will be returned. Otherwise return the pixels in the native format.

Miscellaneous Functions

`cri_GetErrorDescription`

Get error description for the error code. If **`buffer`** is null, return the amount of memory to allocate through the **`buffersize`** parameter. Error description is copied into that string with a null termination. An error is returned if not enough memory is allocated.

`cri_DeallocateMemory`

Deallocate the memory that previously was allocated.

`cri_GetThumbnail`

Get thumbnail image. If **`thumbnail`** is null, do not retrieve this parameter. The routine will allocate the memory. If this function returns a width or height of zero, then the archive does not contain a thumbnail for this image.

`cri_GetSampleIDAndUserComments`

Get sample and user comments. If either buffer is null, return the amount of memory to allocate through the **`buffersize`** parameter. Sample ID and user comments are copied into string with a null termination. An error is returned if not enough memory is allocated.