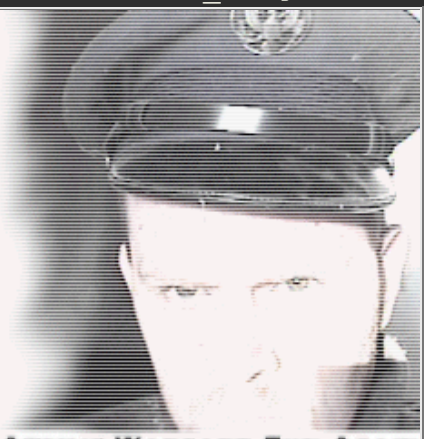
CFX ImageInfoMX



v3.2 October 29th 2004
OPEN SOURCE. FREEWARE
*(includes full **Java** source code)*
Java CFX for **Macromedia** ColdFusionMX and up
by Lewis A. Sellers (aka Min)
http://www.intrafoundation.com
products@intrafoundation.com

Yes, I'm probably for **hire**. Need something? I may or may not have the time, but you can ask.

MAKING ATOMIC WARFARE FUN AGAIN

R E A D T H E V E R S I O N H I S T O R Y

documentation | **version** | **file** | **folder** | **filter** | **extended folder** | **uploadtest** | **bad** | **other** | **unknownfile** -- **stresstest** | **concurrencytest**

NOTE: This tag can return a LOT of information. You may need to set your resolution to 1600x1200 or better to see the entire contents of the tables without scrolling.

F O R E W O R D

This Java language ColdFusion extension tag (that is, a CFX) for **Macromedia** ColdFusionMX, **CFX_ImageInfoMX**, has one simple purpose: It returns information such as width, height, colors, comments, compression type, dpi, frames, etc about graphic files. The file formats it handles include TGA, GIF*, JPG, PNG, BMP, PCX, TIFF, and PSD. Works with all Java versions of ColdFusionMX.

You can either ask it to return information about a single file or about all (known graphic) files in a specified folder. When using the folder option you can also specify a filter mask.

The source code to this program is included as open source. It has been tested under Windows only at the moment. If you have compiled and used it on other platforms please feel free to drop a note.

COPYRIGHT & USAGE RIGHTS

SECURITY ON SHARED SYSTEMS

Is this tag safe to use on a shared system, that is for web hosting or a shared co-located machine? Most Likely. It only reads a file or the files in a folder and returns specific information on known graphic format files. Roughly that makes it about as as safe as allowing a DIR *.jpg,*.gif.

If you have concerns about it's security -- well, you do have the complete source code. If you're paranoid, read through it and recompile a version yourself.

SOURCE CODE

The complete source code is included. It was compiled under Windows 2000 SP4 using Java SDK 1.4 and Netbeans 3.5.1. The version of ColdFusion it was tested against was Redsky ColdFusionMX 6.1.

INSTALLATION

1. Copy CFX_ImageInfoMX.jar to your ColdFusion classes folder. This is usually C:\CFusionMX\cfx\java\ if you're using ColdFusionMX 6 under Windows.
2. Go into the ColdFusion administrator, Extensions, CFX Tags. Register the Java CFX as:

   Tag Name: CFX_ImageInfoMX
   Class Name: CFX_ImageInfoMX

3. If you haven't already, go into the ColdFusion administrator, "Java and JVM" section and add to the Class Path the following:
   `c:\CFusionMX\cfx\java,C:\CFusionMX\lib\cfx.jar,`
   (assuming this is where your version of ColdFusion keeps these files by default.)
4. Reboot. (Or restart ColdFusion server through the services.)

HOW THE TAG WORKS

As stated in the foreword, the purpose of this tag is to return detailed information about image files. You can specify one file at a time or an entire folder. You can also have it use a filter on a folder such as ".jpg", etc.

However you select the files you're interested in, the results are returned in a SQL query called "ImageInfoMX". The query has many fields such as "Width", "Height", etc (see below). Each row is a seperate image file which you can loop through like any SQL query you retrieve from a database.

Additionally, a group of variables are set on each that return basic useful information about it's success and the status of the tag (as well as copyright and version messages).

A SINGLE FILE

```
<CFX_ImageInfoMX FILE="c:\mypictures\mycatssusyandsissy.png">
```

Information about a single file can be returned by using the FILE parameter with the full absolute location of the file on your hard drive (or other local/remote storage media). You should see only 1 or 0 rows returned with a FILE parameter.

AN ENTIRE FOLDER

```
<CFX_ImageInfoMX FOLDER="c:\mypictures\">
```

By specifying a folder location with FOLDER every single image file in the folder will be looked at and information on them gathered up in the query. The number of rows returned could be anywhere from 0 to several thousand depending on how many files are in the folder you point the tag to to begin with.

Use only absolute folder addresses.

USING FILENAME FILTERS

```
<CFX_ImageInfoMX FOLDER="c:\mypictures\" FILTER=".jpg">
```

You can specify an extension to filter the files against. Just tack on the FILTER parm and it will only look at those filenames matching the filter ending.

VARIABLES

The following variables are always returned when the tag is called:

ImageInfoMXDescription
: Text description of the tag and copyright notices.

ImageInfoMXVersion
: The numbered version of the tag. Two part, coma-deliminted. First part is numeric version, second part is string date.

ImageInfoMXFile
: Echo of FILE parameter (only with FILE).

ImageInfoMXFolder
: Echo of FOLDER parameter (only with FOLDER).

ImageInfoMXFilter
: Echo of FILTER parameter (only with FILTER).

ImageInfoMXError
: Displays text error messages if a file could not be opened for some reason and explains why..

QUERY FIELDS

The returned SQL Query "ImageInfoMX" we've mentioned contains the following columns (see below for more detail):

- File
- Path
- Error
- Type
- Format
- LastModified
- Size
- Width
- Height
- Pixels
- PixelBits
- Colors
- ColorType
- Compression
- DPI
- Comments
- Layers
- Duration
- Transparency

- TransparencyIndex
- Format

SUPPORTED FORMATS

The formats currently supported include:

BMP	Microsoft BitMaP (Both Windows and OS/2 versions)
GIF	Compuserve Graphics Interchange Format (GIF87A and GIF89A)
JPG, JPEG	Joint Photographic Experts Group (JPEG/JFIF)
PCX	ZSoft PC Paintbrush
PNG	Portable Network Graphics
PSD	Photoshop Drawing Format
TGA	Truevision Graphics Array Format (a.k.a. TARGA)
TIF, TIFF	Tag Image File Format (TIFF)

THE QUERY FIELDS: DETAILED

The defaults for these fields are either "0" or simply a blank, depending on whether the field is a number or a string. More complicated formats such as TIFF may not always return proper data. Usually you'll get a 0 back in that case, but sometimes it's a very wild number.

Error

A detailed error message. If it's blank then no error occured and the other fields can probably be trusted. Use a simple string length function, ie LEN(Error), in your ImageInfoMX loop to determine if you can trust the data or not.

As of 2.1, formal error codes you can use FIND, etc against have been added. The following error codes are now added to the end of each "sentence" of text describing the error. The text may change from version to version or between image formats, but the following hard error codes will not.

For example, a simple `Find('[CORRUPT]',#Error#)` will allow you to tell if the file was corrupt and display your own custom error message or take your own actions more readily.

The codes are defined thusly:

[GRAMMAR]
: You used the wrong parameters when calling the tag, or mispelled them.

[UNSUPPORTEDFORMAT]
: The file you're trying to get information on is not currently one of those that the tag understands how to get information on. Trying to get info on an AVI would be an example of an unsupported image format.

[UNSUPPORTEDSUBTYPE]
: This means that yes, we do know the image format (for example TIFF), but this particular file is using some rare or new internal format that the tag doesn't know how to completely handle.

[CORRUPT]

The file structure doesn't seem to conform to anything we know about how it should be lain out. This could mean the file is corrupt. It could mean it's not actually the format you think it is (ie, it's a JPG but it's been saved with the BMP extension.) It could also mean it's a radically unsupported image type which has the tag saying "I have no idea how to decode this information".

[ENDOFFILE]

Premature end of file. You'll get this when the tag tries to read a block of information (such as an image header) and unexpectedly the file ends before it should. When you recieve this error you almost always will get a [CORRUPT] error immediately following it for the obvious reason that the image can not be completely and correctly decoded.

[FILEDOESNOTEXIST]

Either the file does not exist or the folder as specified does not.

[SECURITY]

The file probably does exist, but file permission settings are not allowing use to read any of it's data.

File

The file name (without path).

Path

The full path and filename.

Type

The file type, aka the file extension. TGA, GIF, JPG, PNG, BMP, PCX, TIFF and PSD, etc or blank.

LastModification

Last Modification timedate stamp.

Size

Size of the file in bytes.

Width

Image width, in pixels.

Height

Image height, in pixels.

Pixels

Total pixels in an image. Usually width*height.

PixelBits

How many bits are there in a pixel? This is the sum of all channels/components.

Colors

How many colors are used? Should report back up to 63bits worth now (ie, up to 9.22337204e18 colors). Used to report 32bit CMYK formats as 0. Which technically is correct if you understand what 2^32 looks like with a 32bit number. /-) FYI, I'm computing GIF wrong sometimes for unknown reasons.

ColorType

The pixel format: RGB, Grayscale, CLUT, CMYK, etc.

Compression

The compression method used.

DPI

Dots Per Inch (usually 72). With TIFF either some programs seem to store DPI information in "odd" formats or there is some occasional special setting for them I'm not aware of. Anyway, sometimes TIFF DPI's are wrong. Mostly they're right though. I can't say if I just don't understand TIFF 6.0 properly or some graphics programs are munging up the data.

Comments

Comments, if any. Plus other odd text. Some programs seem to want to place binary data in places where only plain text (legally) should be, so "garbage" may appear in the comments of some images.

Layers

Currently GIF only. Number of frames of animation.

Duration

Currently GIF animations only. Milliseconds duration of animation.

Transparency

Transparency. Not currently used.

TransparencyIndex

Returns the index into the color palette for the transparency color. Used only with GIF currently.

Format

The format the image data is stored in. In most cases this is same is the same for each file type, but for more complex image format (such as TIFF) this may vary. The format field also reflects any operating system or platform specifics such as endian-ness or os specific formats. And format revevisions.

Endian-ness, btw, can be either Intel (little endian) or Motorola (big endian). In layman's terms that means PC or Mac format.

Version History

- **3.2**, October 29th 2004.

  Cleaned up documentation, examples and code. Verified will work with Apache 2.

- **3.1**, November 12th 2003.

  Cleaned up code a bit and added better exception handling.

- **3.0b**, November 8th 2003.

  JARed classes. Fixed LastModified bug.

- **3.0a**, November 8th 2003.

  Bug fixes. Added general error variable.

- **3.0**, October 23rd/24th, November 8th 2003.

  Yes, this is a brand new Java version of the old C++ CFX tag. I'd been job-hunting for some steady, long-term work for a change and thought I should get a few of those pesky certificates, ie the SCJP and SCJD. In that spirit converted a few older C++ apps over to Java to brush up on the language before taking the certification exams. Thus is born the Java version of the ancient (circa 1999) CFX_ImageInfo.

  That said, it shouldn't be a surprise if I say that most likely (except for critical bugs) all future update to this tag will happen in the Java version. Maybe.

  I'll get around to properly documenting and cleaning up the source around v3.1 after it's been tested in the wild a while.

Note: Looking for a *Programmer Analyst III* in the **Knoxville/Oak Ridge,TN** area? Email me (at **webmaster@intrafoundation.com**). Long-term, stable and interesting work is preferred, but as always I'm up for short-term contract jobs as well.

Inspirational Music: Radiohead, 2+2=5. *"It's the devil's way now. There is no way out. You can scream and you can shout . It is too late now." (I like songs that reference the tale of Faustus. :))*

- **2.7**, October 3rd 2003.

  Subject: CFX_ImageInfo leaving directory in use?
  Author: Steve Hammonds
  Date/Time: 9/24/2003 11:51:16 AM (231.0)
  I have a template that loops over a directory structure and uses CFX_ImageInfo to retrieve information for the images in the directory. If the image height is greater than the width I cfexecute to irfanview and rotate the image. Then I move the file using cffile and delete the directory. The problem is that I can`t delete the directory after it has been emptied, even manually. I get a sharing violation and I am told that the directory is in use. I have commented out portions of my code and am left to conclude that CFX_ImageInfo is somehow causing Cold Fusion to leave the directory in use. Restarting CF releases the directory and I can delete it. I checked CF administrator and I am am NOT keeping the CFX_ImageInfoMX library loaded. We are running CF 4.5 and the server is Windows 2000 with IIS. Any tips, comments??

Subject: RE:more information
Author: Steve Hammonds
Date/Time: 9/24/2003 11:52:06 AM (232.231)
Here`s some code that can reproduce the problem:

```
<!--- directory strucuture
root
folder1
image file
--->
<cfset
rootdir=`d:\cfusion\template\cfpro\equipment\drawings\test\root2`>
<cfset movedir=`d:\cfusion\template\cfpro\equipment\drawings\test\dest`>
<cfdirectory action=`list` name=`thefolders` directory=`#rootdir#`>
<cfdump var=`#thefolders#`>
<cfloop query=`thefolders`>
<cfif not comparenocase(type,`dir`) and comparenocase(name,`.`) and
comparenocase(name,`..`)>

<CFX_ImageInfo folder=`#rootdir#\#name#\` filter=`*.tif`>
<cfdump var=`#ImageInfo#`>
<cfif len(ImageInfoerror) is 0><!--- got some files --->
<cfset dirname=name>
<cfloop query=`ImageInfo`>
<cffile action=`MOVE` source=`#rootdir#\#dirname#\#file#`
destination=`#movedir#\#file#`>
</cfloop>
<cfdirectory action=`delete` directory=`#rootdir#\#name#`>
</cfif>
</cfif>
</cfloop>
```

Subject: RE:CFX_ImageInfo leaving directory in use?
Author: Steve Hammonds
Date/Time: 9/24/2003 2:35:10 PM (233.231)
Fixed it! The source code was graciously included with the custom tag. In the request.cpp file you make a call to ImageInfoMX. ImageInfoMX declares a HANDLE called hFolder that never gets released. I added `FindClose(hFolder);` to the end of ImageInfoMX function and recompiled. I had to restart the cold Fusion server for it to recognize the new dll file but the problem is solved!

Subject: RE:CFX_ImageInfo leaving directory in use?
Author: Lewis Sellers
Date/Time: 10/2/2003 8:01:31 PM (234.231)
Hello. Been rather busy of late so didn`t seen your post till now. Sorry. But that`s also the thing I love most about open source -- sometimes problems can fix themselves (sort of). :) Anyway, will fix the bug in the version downloadable on this site and add you the credits (like I do everyone that posts a bug fix.) Thanks. Sorry about the hassle though. --min

- **2.6**, July 20th 2003.

  Updated to work under Java CFMX.

NOTE: Was unable to confirm the concurrency tests under CFMX (no updates) as it always kills CFMX dead. (That never happened under CF4 or CF5).

- **2.5**, December 7th 2000.

  Would not report the error of an unsupported file type. Fixed.

- **2.4**, August 2nd, 4th 2000.

  Minor cleanup from last version for bmp and tga, etc.

  Stas, btw, is apparently working on a java-based conversion of this tag. You could almost write it in pure cfm now that cf4.5 has BinaryRead. Almost.

  Note that yes, some field are blank for some image formats. In some cases this is simply because the information doesn't exist in the headers. I'd have to decode the image to find out, and that would slow the tag considerably. Right now it ONLY reads in the headers, ie, framework descriptions, which means it does as little disk accessing and computation as possible. This makes it fairly fast.

  The other reason is that I've simply not gotten around to some of hte fields for some of the formats. Yet.

- **2.3**, August 1st 2000.

  Some minor bug fixes. Also:

  - **Nelson Win** noticed I'd completely forgotten strncpy doesn't append a null. Duh. Fixed.
  - Added BMP O/2 support for Stas (essentially rewrote the bmp module). Submitted CMYKos2 test bmps.
  - Added Format field for clarity

  If I ever have the time I'm going to rewrite a bunch of the code in clean, pure, MFC-free c++. When it was a little tag c was fine, but as it's grown it's showing signs of needing a proper refit to c++.

  Anyone seen my **little gray woods kitty**?

- **2.2**, July 20, 21 2000.

  Added transparency, transparencyindex fields. TransparencyIndex only works for gif currently. I might add it in for other formats later on.

  CONCURRENCY ISSUES:

  Was getting a report the tag was oddly failing in production after a couple hundred iterations of a page and in other odd circumstances. So... I ran some memory leak tests: There weren't any. Ran file handle tests: None were being stranded. Far as I can tell it's pretty much a rock even under high-load (tens of thousands of images) IF you're the only instance using it.

  So, I started looking at concurrency issues. Found a few simple things right off the bat. Sorry about that. Must have forgot to test for it after the 2.0 release. But they still didn't explain the original failure report. So added stresstest.cfm and concurrencytest.cfm.

  Under high concurrent load (many user browsers surfing pages with CFX_ImageInfo calls) the tag was doing very damn odd things, often requiring a cf service restart to gain control of ColdFusion. Thought perhaps it was a file contention issue, so added more error handling and other code in an attempt to isolate and contain any such occurances. Didn't help that much.

I don't entirely understand what's going on (it's probably something simple I'm overlooking that'll dawn on me in a day or two) but the effect is that if two instances of CFX_ImageInfo try to read a file at the same time then the tag is still openable but we read garbage (zero) values. Sometimes it just sits in an infinite loop (waiting for a file?). I'm assuming it's a threading issue, but until it can be isolated the use of CFLOCK is highly suggested (you should have been using it anyway, as it is the general standard with cfx tags.)

Then again it could simply be the machine I'm deving on: CF4.0.1, NT4sp6. All the other deving could have left something unstable. I'll try it on others later. Feel free to do the same and report if you have problems or if you don't. :)

Sorry about the inconvience.

- **2.1**, 6/2/2000.

  This had came up before in a round-about way, and I'd thought about doing such, but I saw a bit of code by Stephane Bisson the other day on the Allaire forums and thought I'd go ahead and take a few minutes to add proper, searchable, error codes: [GRAMMAR], [FILEDOESNOTEXIST], [UNSUPPORTEDFORMAT], [UNSUPPORTEDSUBTYPE], [CORRUPT] and [ENDOFFILE].

  I can see no reason that this should ever change, even if someday we another major upgrade to a 3.x version. Thus, from now on the safe way for error checking is to simply look fo those keywords in the error field.

  Please see the definition for ImageInfoError and ImageInfo.Error for more details.

- **2.0**, 5/28/2000.

  READ THIS! A half year ago (with the 0.9 release) a FOLDER option was added that returned all data back in an SQL query. To retain compatibility with the older version however the FILE parameter still returned the data back as a set of CFSET-compatible variables. As of 2.0 this has finally been dropped. *__ALL__* image information is returned in the SQL query that FOLDER has always used.

  If all you've ever used was FOLDER then this doesn't affect you. If any of your code uses FILE however you will need to modify your cfm pages to reflect this change or continue using the old 1.8 version. However, in most cases, changing your code will be fairly quick and painless. After all, how long does it take to change "ImageInfoWidth" to "ImageInfo.Width", etc? :-)

  This change comes about because the structure of the source code was rewritten. It's now harder for novices to read, but it's more in line with proper coding practices. (It had been nagging at me for the last few months).

  Anyway, what's done is done. Hopefully the code is tighter, faster and less problematic as a result.

  Additional changes include:

  - Fix of repetitive "premature eof" error messages. You should only see one per file now.
  - Fixed major end-of-file bug for JPG. Should process them faster now as well.
  - Added "path" to the returned query. It contains the full path and filename as "file" used to.
  - Changed "file" to only return the filename and not the entire path.
  - Added FILTER for the FOLDER option. You can now use DOS wildcard filter characters (* and ?).
  - misc cleanup
  - fixed bug that could leave out first row in the returned query.
  - Removed fpx (flashpix) stub. If you have header info for fpx, send it. I have no idea what the format for fpx is.

- Didn't mention it before, but, while I'm writing a novella-length update, I believe Kristin Aileen Motlagh has a book coming out with this tag in it. (At least, I recall signing a release form for something a couple months back. hrmmm...)
- tighter code and less file accessing ***should*** made the tag faster than it ever has been

NOTE: In response to some confusion, no, this tag does not have any relation to the CFX_ImgInfo program by Scott Howard circa 1997. I was unware of the tag in question until sometime after I created this one, otherwise I probably would have given it another name (to avoid people asking me questions about Scott's tag.)

- **1.8**, 4/14/2000. Misc. Fixed PNG comments and comments in general. Some programs create PNG Creation Time text that is not to spec.
- **1.7**, 4/7/2000. Fix LastMod/Created. Add LastAccess.
- **1.6**, Quick change for g. butterwegge so that LastMod and Created fields use ODBC compatible timedate format.
  Zipped all test images as a seperate .ZIP so you don't have to download them over and over.
- **1.5**, 3/28/2000. By request (g. butterwegge) added LastMod and Created fields. Misc other.
- **1.4**, 2/28/2000. Documentation amplifications and addition of a more "formal" copyright statement. No significant change to the code base itself. (As much as I might be tempted.)
- **1.3**, 2/7/2000. Apparently Drew was having problems getting the tag to work with mapped shares (network drives). Reworked the code a little for shares. Also made error messages a little more verbose.
- **1.2**, 1/21/2000. bug fix. GIF's with a color resolution other than "3" (less than 256 colors) were being incorrectly read. Fixed according to/for *Golem*.
- **1.1**, 1/18/2000. There were requests for jpeg DPI info (phillippi) and reporting of gif animation features such as frames/layers and duration (golem). Those have been added. There probably will be a cleaned-up 1.2 shortly as I'm about to release the commercial big brother to this tag. Since it's a c++ dcom object (cf or asp) it doesn't exactly translate easily to the simplier c code of this tag, but a few new features will probably find their way into it.
- **1.0 Stable**, 12/15/1999. Well, it's been over half a year since the tag was released into the wild, and there doesn't seem to be any currently existant issues with the code. So... I went ahead and graduated it to a 1.0 gamma version. Congrats CFX_ImageInfoMX. /-) From now on we're in a feature-lock. That means only issues to make the tag faster and more stable will be addressed. I need to read up on exactly what the GPL license specifically involves before I officially state it, but this tag and it's included code are essentially GPL'd in all but name.
- **.13**. 12/3/1999,12/15/1999. More final clean up. Feature lock. Copied code-base to CFX_ImageInfoMXSM/CFX_Thumbnails project (generates thumbnails). Stable. Should be last mod for at least the next half year. Added hooks for a couple a/v formats (AVI,MOV,MPEG) but no active code. Something for the future perhaps. Moving future additions to CFX_ImageFX.
- **.12** 12/2/1999. Bug hunt and polishing.
- **.11**, 11/30/1999. Added TIFF support. Sort of. TIFF is a hideously complex image format with some fairly silly concepts of flow design. More or less works though.
  Fixed other misc problems like reporting >=32 bit color formats. Added PSD 2.0.
- **.10**, 11/30/1999. Limited release. Fixed problem with JPG comments not having nulls. Fixed DPI always reporting 72. And other misc.
- **.09**, 11/29/1999. **Mark Gaulin** pointed out I'd forgotten a free() in a function. fixed. Recoded many portions such as bmp.h, and tweaked them to report information a little more accurately in some cases. Added FOLDER option. Added code to read PCX.
- **.08**, Unreleased
- **.07**, 5/21/99 The long-standing release version
- **.06** and before, prehistoric

*No, at no time was LZW code harmed in the making of this tag. It reads through generic header information only. Look for yourself.