

Photo Finish

1

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>File Index</b>	<b>9</b>
4.1	File List . . . . .	9
<b>5</b>	<b>Namespace Documentation</b>	<b>11</b>
5.1	CMS Namespace Reference . . . . .	11
5.1.1	Enumeration Type Documentation . . . . .	12
5.1.1.1	ColourModel . . . . .	12
5.1.1.2	Intent . . . . .	13
5.1.2	Function Documentation . . . . .	13
5.1.2.1	istream_close() . . . . .	13
5.1.2.2	istream_read() . . . . .	14
5.1.2.3	istream_seek() . . . . .	14
5.1.2.4	istream_tell() . . . . .	14
5.1.2.5	istream_write() . . . . .	14
5.1.2.6	OpenIOHandlerFromIFStream() . . . . .	14
5.1.2.7	OpenIOHandlerFromIStream() . . . . .	15

5.1.2.8	<code>operator&lt;&lt;()</code> [1/2]	15
5.1.2.9	<code>operator&lt;&lt;()</code> [2/2]	15
5.1.2.10	<code>ostream_close()</code>	15
5.1.2.11	<code>ostream_read()</code>	15
5.1.2.12	<code>ostream_seek()</code>	16
5.1.2.13	<code>ostream_tell()</code>	16
5.1.2.14	<code>ostream_write()</code>	16
5.2	PhotoFinish Namespace Reference	16
5.2.1	Typedef Documentation	21
5.2.1.1	<code>hash</code>	21
5.2.1.2	<code>jxr_format_subst</code>	21
5.2.1.3	<code>multihash</code>	22
5.2.1.4	<code>rulerlist</code>	22
5.2.1.5	<code>rulerpair</code>	22
5.2.1.6	<code>stringlist</code>	22
5.2.1.7	<code>subst_table</code>	22
5.2.2	Function Documentation	22
5.2.2.1	<code>add_ruler_pins()</code>	23
5.2.2.2	<code>add_rulers()</code>	23
5.2.2.3	<code>closest_Rational()</code>	23
5.2.2.4	<code>copy_le_to()</code>	23
5.2.2.5	<code>error_callback()</code>	24
5.2.2.6	<code>exif_key_read()</code>	24
5.2.2.7	<code>exif_value_read()</code>	24
5.2.2.8	<code>exists()</code>	24
5.2.2.9	<code>info_callback()</code>	24
5.2.2.10	<code>iptc_key_read()</code>	25
5.2.2.11	<code>jpeg_error_exit()</code>	25
5.2.2.12	<code>jpeg_istream_fill_input_buffer()</code>	25
5.2.2.13	<code>jpeg_istream_init_source()</code>	25

5.2.2.14	<a href="#">jpeg_istream_resync_to_restart()</a>	25
5.2.2.15	<a href="#">jpeg_istream_skip_input_data()</a>	26
5.2.2.16	<a href="#">jpeg_istream_src()</a>	26
5.2.2.17	<a href="#">jpeg_istream_src_free()</a>	26
5.2.2.18	<a href="#">jpeg_istream_term_source()</a>	26
5.2.2.19	<a href="#">jpeg_ostream_dest()</a>	27
5.2.2.20	<a href="#">jpeg_ostream_dest_free()</a>	27
5.2.2.21	<a href="#">jpeg_read_profile()</a>	27
5.2.2.22	<a href="#">jpeg_write_profile()</a>	27
5.2.2.23	<a href="#">jpegfile_scan_grayscale()</a>	28
5.2.2.24	<a href="#">jpegfile_scan_RGB()</a>	28
5.2.2.25	<a href="#">jxr_cms_format()</a>	28
5.2.2.26	<a href="#">jxr_pixel_format()</a>	28
5.2.2.27	<a href="#">last_write_time()</a>	28
5.2.2.28	<a href="#">lcms2_error_adaptor()</a>	29
5.2.2.29	<a href="#">lcms2_errorhandler()</a>	29
5.2.2.30	<a href="#">limitval()</a>	29
5.2.2.31	<a href="#">limitval&lt; double &gt;()</a>	29
5.2.2.32	<a href="#">limitval&lt; float &gt;()</a>	29
5.2.2.33	<a href="#">limitval&lt; unsigned char &gt;()</a>	30
5.2.2.34	<a href="#">limitval&lt; unsigned int &gt;()</a>	30
5.2.2.35	<a href="#">limitval&lt; unsigned long long &gt;()</a>	30
5.2.2.36	<a href="#">limitval&lt; unsigned short int &gt;()</a>	30
5.2.2.37	<a href="#">operator&lt;&lt;()</a>	30
5.2.2.38	<a href="#">parse_Rational()</a>	31
5.2.2.39	<a href="#">png_end_cb()</a>	31
5.2.2.40	<a href="#">png_flush_ostream_cb()</a>	31
5.2.2.41	<a href="#">png_info_cb()</a>	31
5.2.2.42	<a href="#">png_row_cb()</a>	32
5.2.2.43	<a href="#">png_write_ostream_cb()</a>	32

5.2.2.44	<code>profile_name()</code>	32
5.2.2.45	<code>read_le32()</code>	32
5.2.2.46	<code>read_planar()</code>	33
5.2.2.47	<code>scaleval()</code>	33
5.2.2.48	<code>scaleval&lt; double &gt;()</code>	33
5.2.2.49	<code>scaleval&lt; float &gt;()</code>	33
5.2.2.50	<code>scaleval&lt; unsigned char &gt;()</code>	33
5.2.2.51	<code>scaleval&lt; unsigned int &gt;()</code>	34
5.2.2.52	<code>scaleval&lt; unsigned long long &gt;()</code>	34
5.2.2.53	<code>scaleval&lt; unsigned short int &gt;()</code>	34
5.2.2.54	<code>transfer_alpha()</code>	34
5.2.2.55	<code>transfer_alpha_typed()</code>	34
5.2.2.56	<code>transfer_alpha_typed2()</code>	35
5.2.2.57	<code>warning_callback()</code>	35
5.2.2.58	<code>webp_stream_writer_func()</code>	35
5.2.2.59	<code>write_be()</code>	35
5.2.2.60	<code>write_packed()</code>	36
5.2.2.61	<code>write_planar()</code>	36
5.2.2.62	<code>xmp_key_read()</code>	36
5.2.3	Variable Documentation	36
5.2.3.1	<code>benchmark_mode</code>	36
5.2.3.2	<code>EXIF_key_subst</code>	37
5.2.3.3	<code>EXIF_value_subst</code>	37
5.2.3.4	<code>header</code>	37
5.2.3.5	<code>IPTC_key_subst</code>	37
5.2.3.6	<code>JXR_format_table</code>	38
5.2.3.7	<code>WebP_presets</code>	38
5.2.3.8	<code>XMP_key_subst</code>	38

<b>6 Class Documentation</b>	<b>39</b>
6.1 PhotoFinish::cmsTypeError Class Reference	39
6.1.1 Detailed Description	39
6.1.2 Constructor & Destructor Documentation	39
6.1.2.1 cmsTypeError()	39
6.1.3 Member Function Documentation	40
6.1.3.1 what()	40
6.2 PhotoFinish::CropSolver Class Reference	40
6.2.1 Detailed Description	40
6.2.2 Constructor & Destructor Documentation	40
6.2.2.1 CropSolver()	41
6.2.3 Member Function Documentation	41
6.2.3.1 solve()	41
6.3 PhotoFinish::D_JP2 Class Reference	41
6.3.1 Detailed Description	42
6.3.2 Constructor & Destructor Documentation	42
6.3.2.1 D_JP2()	42
6.3.3 Member Function Documentation	42
6.3.3.1 add_variables()	43
6.3.3.2 num_qualities()	43
6.3.3.3 num_rates()	43
6.3.3.4 numresolutions()	43
6.3.3.5 prog_order()	43
6.3.3.6 quality()	44
6.3.3.7 rate()	44
6.3.3.8 read_config()	44
6.3.3.9 reversible()	44
6.3.3.10 set_irreversible()	44
6.3.3.11 set_numresolutions()	45
6.3.3.12 set_prog_order()	45

6.3.3.13	<a href="#">set_qualities()</a>	45
6.3.3.14	<a href="#">set_quality()</a>	45
6.3.3.15	<a href="#">set_rate()</a>	45
6.3.3.16	<a href="#">set_rates()</a>	46
6.3.3.17	<a href="#">set_reversible()</a>	46
6.3.3.18	<a href="#">set_tile_size()</a>	46
6.3.3.19	<a href="#">tile_size()</a>	46
6.4	<a href="#">PhotoFinish::D_JPEG Class Reference</a>	46
6.4.1	<a href="#">Detailed Description</a>	47
6.4.2	<a href="#">Constructor &amp; Destructor Documentation</a>	47
6.4.2.1	<a href="#">D_JPEG() [1/2]</a>	47
6.4.2.2	<a href="#">D_JPEG() [2/2]</a>	47
6.4.3	<a href="#">Member Function Documentation</a>	48
6.4.3.1	<a href="#">add_variables()</a>	48
6.4.3.2	<a href="#">progressive()</a>	48
6.4.3.3	<a href="#">quality()</a>	48
6.4.3.4	<a href="#">read_config()</a>	48
6.4.3.5	<a href="#">sample()</a>	49
6.4.3.6	<a href="#">set_progressive()</a>	49
6.4.3.7	<a href="#">set_quality()</a>	49
6.4.3.8	<a href="#">set_sample()</a>	49
6.5	<a href="#">PhotoFinish::D_JXR Class Reference</a>	49
6.5.1	<a href="#">Detailed Description</a>	50
6.5.2	<a href="#">Constructor &amp; Destructor Documentation</a>	50
6.5.2.1	<a href="#">D_JXR()</a>	50
6.5.3	<a href="#">Member Function Documentation</a>	50
6.5.3.1	<a href="#">add_variables()</a>	51
6.5.3.2	<a href="#">alphaq()</a>	51
6.5.3.3	<a href="#">overlap()</a>	51
6.5.3.4	<a href="#">progressive()</a>	51



6.5.3.5	<a href="#">quality()</a>	51
6.5.3.6	<a href="#">read_config()</a>	52
6.5.3.7	<a href="#">set_alphaq()</a>	52
6.5.3.8	<a href="#">set_overlap()</a>	52
6.5.3.9	<a href="#">set_progressive()</a>	52
6.5.3.10	<a href="#">set_quality()</a>	52
6.5.3.11	<a href="#">set_sequential()</a>	52
6.5.3.12	<a href="#">set_subsampling()</a>	53
6.5.3.13	<a href="#">set_tilesize()</a>	53
6.5.3.14	<a href="#">subsampling()</a>	53
6.5.3.15	<a href="#">tilesize()</a>	53
6.6	<a href="#">PhotoFinish::D_PNG Class Reference</a>	53
6.6.1	<a href="#">Detailed Description</a>	54
6.6.2	<a href="#">Constructor &amp; Destructor Documentation</a>	54
6.6.2.1	<a href="#">D_PNG()</a>	54
6.6.3	<a href="#">Member Function Documentation</a>	54
6.6.3.1	<a href="#">read_config()</a>	54
6.7	<a href="#">PhotoFinish::D_profile Class Reference</a>	54
6.7.1	<a href="#">Detailed Description</a>	55
6.7.2	<a href="#">Member Typedef Documentation</a>	55
6.7.2.1	<a href="#">ptr</a>	55
6.7.3	<a href="#">Constructor &amp; Destructor Documentation</a>	56
6.7.3.1	<a href="#">D_profile()</a> [1/4]	56
6.7.3.2	<a href="#">D_profile()</a> [2/4]	56
6.7.3.3	<a href="#">D_profile()</a> [3/4]	56
6.7.3.4	<a href="#">D_profile()</a> [4/4]	56
6.7.3.5	<a href="#">~D_profile()</a>	57
6.7.4	<a href="#">Member Function Documentation</a>	57
6.7.4.1	<a href="#">data()</a>	57
6.7.4.2	<a href="#">data_size()</a>	57

6.7.4.3	filepath()	57
6.7.4.4	has_data()	58
6.7.4.5	name()	58
6.7.4.6	operator=()	58
6.7.4.7	profile()	58
6.7.4.8	read_config()	58
6.8	PhotoFinish::D_resize Class Reference	59
6.8.1	Detailed Description	59
6.8.2	Constructor & Destructor Documentation	59
6.8.2.1	D_resize()	59
6.8.3	Member Function Documentation	60
6.8.3.1	filter()	60
6.8.3.2	lanczos()	60
6.8.3.3	read_config()	60
6.8.3.4	support()	60
6.9	PhotoFinish::D_sharpen Class Reference	61
6.9.1	Detailed Description	61
6.9.2	Constructor & Destructor Documentation	61
6.9.2.1	D_sharpen()	61
6.9.3	Member Function Documentation	61
6.9.3.1	radius()	62
6.9.3.2	read_config()	62
6.9.3.3	sigma()	62
6.10	PhotoFinish::D_target Class Reference	62
6.10.1	Detailed Description	63
6.10.2	Member Typedef Documentation	63
6.10.2.1	ptr	63
6.10.3	Constructor & Destructor Documentation	63
6.10.3.1	D_target() [1/2]	63
6.10.3.2	D_target() [2/2]	64

6.10.4	Member Function Documentation	64
6.10.4.1	height()	64
6.10.4.2	name()	64
6.10.4.3	read_config()	64
6.10.4.4	size()	64
6.10.4.5	width()	65
6.10.5	Member Data Documentation	65
6.10.5.1	_height	65
6.10.5.2	_name	65
6.10.5.3	_size	65
6.10.5.4	_width	65
6.11	PhotoFinish::D_thumbnail Class Reference	66
6.11.1	Detailed Description	66
6.11.2	Constructor & Destructor Documentation	66
6.11.2.1	D_thumbnail()	66
6.11.3	Member Function Documentation	66
6.11.3.1	generate()	67
6.11.3.2	maxheight()	67
6.11.3.3	maxwidth()	67
6.11.3.4	read_config()	67
6.12	PhotoFinish::D_TIFF Class Reference	67
6.12.1	Detailed Description	68
6.12.2	Constructor & Destructor Documentation	68
6.12.2.1	D_TIFF() [1/2]	68
6.12.2.2	D_TIFF() [2/2]	68
6.12.3	Member Function Documentation	69
6.12.3.1	add_variables()	69
6.12.3.2	artist()	69
6.12.3.3	compression()	69
6.12.3.4	copyright()	69

6.12.3.5	<a href="#">read_config()</a>	70
6.12.3.6	<a href="#">set_artist()</a>	70
6.12.3.7	<a href="#">set_compression()</a>	70
6.12.3.8	<a href="#">set_copyright()</a>	70
6.13	<a href="#">PhotoFinish::D_WebP Class Reference</a>	70
6.13.1	<a href="#">Detailed Description</a>	71
6.13.2	<a href="#">Constructor &amp; Destructor Documentation</a>	71
6.13.2.1	<a href="#">D_WebP()</a>	71
6.13.3	<a href="#">Member Function Documentation</a>	71
6.13.3.1	<a href="#">add_variables()</a>	71
6.13.3.2	<a href="#">lossless()</a>	72
6.13.3.3	<a href="#">lossy()</a>	72
6.13.3.4	<a href="#">method()</a>	72
6.13.3.5	<a href="#">preset()</a>	72
6.13.3.6	<a href="#">quality()</a>	72
6.13.3.7	<a href="#">read_config()</a>	72
6.13.3.8	<a href="#">set_lossless()</a>	73
6.13.3.9	<a href="#">set_lossy()</a>	73
6.13.3.10	<a href="#">set_method()</a>	73
6.13.3.11	<a href="#">set_preset()</a>	73
6.13.3.12	<a href="#">set_quality()</a>	73
6.14	<a href="#">PhotoFinish::definable&lt; T &gt; Class Template Reference</a>	74
6.14.1	<a href="#">Detailed Description</a>	74
6.14.2	<a href="#">Constructor &amp; Destructor Documentation</a>	74
6.14.2.1	<a href="#">definable() [1/2]</a>	75
6.14.2.2	<a href="#">definable() [2/2]</a>	75
6.14.3	<a href="#">Member Function Documentation</a>	75
6.14.3.1	<a href="#">defined()</a>	75
6.14.3.2	<a href="#">get() [1/2]</a>	75
6.14.3.3	<a href="#">get() [2/2]</a>	76

6.14.3.4	<a href="#">operator T()</a>	76
6.14.3.5	<a href="#">operator-&gt;() [1/2]</a>	76
6.14.3.6	<a href="#">operator-&gt;() [2/2]</a>	76
6.14.3.7	<a href="#">operator=()</a>	77
6.14.3.8	<a href="#">set_defined()</a>	77
6.14.3.9	<a href="#">undefine()</a>	77
6.14.4	<a href="#">Friends And Related Function Documentation</a>	77
6.14.4.1	<a href="#">operator&lt;&lt;</a>	77
6.15	<a href="#">PhotoFinish::Destination Class Reference</a>	78
6.15.1	<a href="#">Detailed Description</a>	79
6.15.2	<a href="#">Member Typedef Documentation</a>	79
6.15.2.1	<a href="#">ptr</a>	79
6.15.3	<a href="#">Constructor &amp; Destructor Documentation</a>	79
6.15.3.1	<a href="#">Destination() [1/2]</a>	79
6.15.3.2	<a href="#">Destination() [2/2]</a>	80
6.15.3.3	<a href="#">~Destination()</a>	80
6.15.4	<a href="#">Member Function Documentation</a>	80
6.15.4.1	<a href="#">add_variables()</a>	80
6.15.4.2	<a href="#">best_frame()</a>	80
6.15.4.3	<a href="#">clear_profile()</a>	81
6.15.4.4	<a href="#">depth()</a>	81
6.15.4.5	<a href="#">dir()</a>	81
6.15.4.6	<a href="#">dupe()</a>	81
6.15.4.7	<a href="#">forcegrey()</a>	81
6.15.4.8	<a href="#">forcergb()</a>	82
6.15.4.9	<a href="#">format()</a>	82
6.15.4.10	<a href="#">get_profile()</a>	82
6.15.4.11	<a href="#">has_targets()</a>	82
6.15.4.12	<a href="#">intent()</a>	82
6.15.4.13	<a href="#">jp2()</a>	83

6.15.4.14 jpeg()	83
6.15.4.15 jxr()	83
6.15.4.16 modify_format()	83
6.15.4.17 name()	83
6.15.4.18 noresize()	84
6.15.4.19 num_targets()	84
6.15.4.20 operator=()	84
6.15.4.21 png()	84
6.15.4.22 profile()	84
6.15.4.23 read_config()	85
6.15.4.24 resize()	85
6.15.4.25 set_depth()	85
6.15.4.26 set_jp2()	85
6.15.4.27 set_jpeg()	85
6.15.4.28 set_jxr()	86
6.15.4.29 set_png()	86
6.15.4.30 set_profile() [1/2]	86
6.15.4.31 set_profile() [2/2]	86
6.15.4.32 set_tiff()	86
6.15.4.33 set_webp()	87
6.15.4.34 sharpen()	87
6.15.4.35 size()	87
6.15.4.36 targets()	87
6.15.4.37 thumbnail()	87
6.15.4.38 tiff()	87
6.15.4.39 webp()	88
6.16 PhotoFinish::DestinationError Class Reference	88
6.16.1 Detailed Description	88
6.16.2 Constructor & Destructor Documentation	88
6.16.2.1 DestinationError()	88

6.16.3	Member Function Documentation	89
6.16.3.1	what()	89
6.17	PhotoFinish::Destinations Class Reference	89
6.17.1	Detailed Description	90
6.17.2	Member Typedef Documentation	90
6.17.2.1	const_iterator	90
6.17.2.2	iterator	90
6.17.3	Constructor & Destructor Documentation	90
6.17.3.1	Destinations() [1/2]	90
6.17.3.2	Destinations() [2/2]	91
6.17.3.3	~Destinations()	91
6.17.4	Member Function Documentation	91
6.17.4.1	begin() [1/2]	91
6.17.4.2	begin() [2/2]	91
6.17.4.3	count()	91
6.17.4.4	end() [1/2]	92
6.17.4.5	end() [2/2]	92
6.17.4.6	Load()	92
6.17.4.7	operator=()	92
6.17.4.8	operator[]()	92
6.17.5	Friends And Related Function Documentation	92
6.17.5.1	begin	93
6.17.5.2	end	93
6.18	PhotoFinish::Ditherer Class Reference	93
6.18.1	Detailed Description	93
6.18.2	Constructor & Destructor Documentation	94
6.18.2.1	Ditherer()	94
6.18.2.2	~Ditherer()	94
6.18.3	Member Function Documentation	94
6.18.3.1	dither()	94

6.18.4	Member Data Documentation . . . . .	95
6.18.4.1	cmsBaseType . . . . .	95
6.19	PhotoFinish::ErrorMsg Class Reference . . . . .	95
6.19.1	Detailed Description . . . . .	96
6.19.2	Constructor & Destructor Documentation . . . . .	96
6.19.2.1	ErrorMsg() . . . . .	96
6.19.3	Member Function Documentation . . . . .	96
6.19.3.1	what() . . . . .	96
6.19.4	Member Data Documentation . . . . .	96
6.19.4.1	_msg . . . . .	97
6.20	PhotoFinish::FileContentError Class Reference . . . . .	97
6.20.1	Detailed Description . . . . .	97
6.20.2	Constructor & Destructor Documentation . . . . .	97
6.20.2.1	FileContentError() [1/2] . . . . .	97
6.20.2.2	FileContentError() [2/2] . . . . .	98
6.20.3	Member Function Documentation . . . . .	98
6.20.3.1	what() . . . . .	98
6.21	PhotoFinish::FileError Class Reference . . . . .	98
6.21.1	Detailed Description . . . . .	99
6.21.2	Constructor & Destructor Documentation . . . . .	99
6.21.2.1	FileError() [1/2] . . . . .	99
6.21.2.2	FileError() [2/2] . . . . .	100
6.21.3	Member Function Documentation . . . . .	100
6.21.3.1	what() . . . . .	100
6.21.4	Member Data Documentation . . . . .	100
6.21.4.1	_filepath . . . . .	100
6.22	PhotoFinish::FileOpenError Class Reference . . . . .	101
6.22.1	Detailed Description . . . . .	101
6.22.2	Constructor & Destructor Documentation . . . . .	101
6.22.2.1	FileOpenError() [1/2] . . . . .	101



6.22.2.2	<a href="#">FileOpenError()</a> [ 2 / 2 ]	102
6.22.3	<a href="#">Member Function Documentation</a>	102
6.22.3.1	<a href="#">what()</a>	102
6.23	<a href="#">CMS::Format Class Reference</a>	102
6.23.1	<a href="#">Detailed Description</a>	105
6.23.2	<a href="#">Constructor &amp; Destructor Documentation</a>	105
6.23.2.1	<a href="#">Format()</a>	105
6.23.3	<a href="#">Member Function Documentation</a>	105
6.23.3.1	<a href="#">bytes_per_channel()</a>	106
6.23.3.2	<a href="#">bytes_per_pixel()</a>	106
6.23.3.3	<a href="#">channels()</a>	106
6.23.3.4	<a href="#">CMYK8()</a>	106
6.23.3.5	<a href="#">colour_model()</a>	106
6.23.3.6	<a href="#">extra_channels()</a>	107
6.23.3.7	<a href="#">Grey16()</a>	107
6.23.3.8	<a href="#">Grey8()</a>	107
6.23.3.9	<a href="#">is_16bit()</a>	107
6.23.3.10	<a href="#">is_32bit()</a>	107
6.23.3.11	<a href="#">is_8bit()</a>	108
6.23.3.12	<a href="#">is_chocolate()</a>	108
6.23.3.13	<a href="#">is_double()</a>	108
6.23.3.14	<a href="#">is_endianswapped()</a>	108
6.23.3.15	<a href="#">is_float()</a>	108
6.23.3.16	<a href="#">is_fp()</a>	109
6.23.3.17	<a href="#">is_half()</a>	109
6.23.3.18	<a href="#">is_integer()</a>	109
6.23.3.19	<a href="#">is_optimised()</a>	109
6.23.3.20	<a href="#">is_packed()</a>	109
6.23.3.21	<a href="#">is_planar()</a>	110
6.23.3.22	<a href="#">is_premult_alpha()</a>	110

6.23.3.23 is_swapped()	110
6.23.3.24 is_swappedfirst()	110
6.23.3.25 is_vanilla()	110
6.23.3.26 LabDouble()	111
6.23.3.27 LabFloat()	111
6.23.3.28 operator cmsUInt32Number()	111
6.23.3.29 RGB16()	111
6.23.3.30 RGB8()	111
6.23.3.31 scaleval()	112
6.23.3.32 set_16bit()	112
6.23.3.33 set_32bit()	112
6.23.3.34 set_8bit()	112
6.23.3.35 set_channel_type() [1/8]	112
6.23.3.36 set_channel_type() [2/8]	113
6.23.3.37 set_channel_type() [3/8]	113
6.23.3.38 set_channel_type() [4/8]	113
6.23.3.39 set_channel_type() [5/8]	113
6.23.3.40 set_channel_type() [6/8]	113
6.23.3.41 set_channel_type() [7/8]	114
6.23.3.42 set_channel_type() [8/8]	114
6.23.3.43 set_chocolate()	114
6.23.3.44 set_colour_model()	114
6.23.3.45 set_double()	114
6.23.3.46 set_endianswap()	115
6.23.3.47 set_extra_channels()	115
6.23.3.48 set_float()	115
6.23.3.49 set_half()	115
6.23.3.50 set_packed()	115
6.23.3.51 set_planar()	116
6.23.3.52 set_premult_alpha()	116

6.23.3.53	<a href="#">set_swap()</a>	116
6.23.3.54	<a href="#">set_swapfirst()</a>	116
6.23.3.55	<a href="#">set_vanilla()</a>	116
6.23.3.56	<a href="#">total_channels()</a>	117
6.23.3.57	<a href="#">unset_endianswap()</a>	117
6.23.3.58	<a href="#">unset_premult_alpha()</a>	117
6.23.3.59	<a href="#">unset_swap()</a>	117
6.23.3.60	<a href="#">unset_swapfirst()</a>	117
6.23.4	<a href="#">Friends And Related Function Documentation</a>	117
6.23.4.1	<a href="#">Transform</a>	118
6.24	<a href="#">PhotoFinish::Frame Class Reference</a>	118
6.24.1	<a href="#">Detailed Description</a>	119
6.24.2	<a href="#">Member Typedef Documentation</a>	119
6.24.2.1	<a href="#">ptr</a>	119
6.24.3	<a href="#">Constructor &amp; Destructor Documentation</a>	119
6.24.3.1	<a href="#">Frame() [1/2]</a>	119
6.24.3.2	<a href="#">Frame() [2/2]</a>	120
6.24.4	<a href="#">Member Function Documentation</a>	120
6.24.4.1	<a href="#">crop_h()</a>	120
6.24.4.2	<a href="#">crop_resize()</a>	120
6.24.4.3	<a href="#">crop_w()</a>	121
6.24.4.4	<a href="#">crop_x()</a>	121
6.24.4.5	<a href="#">crop_y()</a>	121
6.24.4.6	<a href="#">waste()</a>	121
6.25	<a href="#">PhotoFinish::GaussianSharpen Class Reference</a>	122
6.25.1	<a href="#">Detailed Description</a>	122
6.25.2	<a href="#">Constructor &amp; Destructor Documentation</a>	122
6.25.2.1	<a href="#">GaussianSharpen() [1/2]</a>	122
6.25.2.2	<a href="#">GaussianSharpen() [2/2]</a>	122
6.26	<a href="#">PhotoFinish::Image Class Reference</a>	123

6.26.1 Detailed Description . . . . .	124
6.26.2 Member Typedef Documentation . . . . .	125
6.26.2.1 ptr . . . . .	125
6.26.3 Constructor & Destructor Documentation . . . . .	125
6.26.3.1 Image() . . . . .	125
6.26.3.2 ~Image() . . . . .	125
6.26.4 Member Function Documentation . . . . .	125
6.26.4.1 alpha_mult() . . . . .	126
6.26.4.2 at() [1/2] . . . . .	127
6.26.4.3 at() [2/2] . . . . .	127
6.26.4.4 check_rowdata_alloc() . . . . .	127
6.26.4.5 default_profile() [1/2] . . . . .	127
6.26.4.6 default_profile() [2/2] . . . . .	128
6.26.4.7 EXIFtags() . . . . .	128
6.26.4.8 format() . . . . .	128
6.26.4.9 free_row() . . . . .	128
6.26.4.10 has_profile() . . . . .	128
6.26.4.11 height() . . . . .	129
6.26.4.12 IPTCtags() . . . . .	129
6.26.4.13 pixel_size() . . . . .	129
6.26.4.14 profile() . . . . .	129
6.26.4.15 row() . . . . .	129
6.26.4.16 row_size() . . . . .	130
6.26.4.17 set_profile() . . . . .	130
6.26.4.18 set_resolution() [1/2] . . . . .	130
6.26.4.19 set_resolution() [2/2] . . . . .	130
6.26.4.20 set_resolution_from_size() . . . . .	130
6.26.4.21 set_xres() . . . . .	131
6.26.4.22 set_yres() . . . . .	131
6.26.4.23 transform_colour() . . . . .	131

6.26.4.24 transform_colour_inplace()	131
6.26.4.25 un_alpha_mult()	132
6.26.4.26 width()	132
6.26.4.27 XMPTags()	132
6.26.4.28 xres()	133
6.26.4.29 yres()	133
6.27 PhotoFinish::ImageFilepath Class Reference	133
6.27.1 Detailed Description	134
6.27.2 Constructor & Destructor Documentation	134
6.27.2.1 ImageFilepath() [1/2]	134
6.27.2.2 ImageFilepath() [2/2]	134
6.27.3 Member Function Documentation	134
6.27.3.1 filepath()	135
6.27.3.2 fix_filepath()	135
6.27.3.3 fixed_filepath()	135
6.27.3.4 format()	135
6.27.4 Friends And Related Function Documentation	135
6.27.4.1 operator<<	135
6.28 PhotoFinish::ImageReader Class Reference	136
6.28.1 Detailed Description	136
6.28.2 Member Typedef Documentation	136
6.28.2.1 ptr	137
6.28.3 Constructor & Destructor Documentation	137
6.28.3.1 ImageReader()	137
6.28.4 Member Function Documentation	137
6.28.4.1 extract_tags()	137
6.28.4.2 open()	137
6.28.4.3 read() [1/2]	138
6.28.4.4 read() [2/2]	138
6.28.5 Member Data Documentation	138

6.28.5.1	<code>_filepath</code>	138
6.28.5.2	<code>_is_open</code>	139
6.29	PhotoFinish::ImageWriter Class Reference	139
6.29.1	Detailed Description	140
6.29.2	Member Typedef Documentation	140
6.29.2.1	<code>ptr</code>	140
6.29.3	Constructor & Destructor Documentation	140
6.29.3.1	<code>ImageWriter()</code>	140
6.29.4	Member Function Documentation	140
6.29.4.1	<code>add_variables()</code>	141
6.29.4.2	<code>embed_tags()</code>	141
6.29.4.3	<code>open()</code>	141
6.29.4.4	<code>preferred_format()</code>	141
6.29.4.5	<code>write()</code>	142
6.29.5	Member Data Documentation	142
6.29.5.1	<code>_filepath</code>	142
6.29.5.2	<code>_is_open</code>	142
6.30	PhotoFinish::jpeg_destination_state_t Struct Reference	142
6.30.1	Detailed Description	143
6.30.2	Member Data Documentation	143
6.30.2.1	<code>buffer</code>	143
6.30.2.2	<code>buffer_size</code>	143
6.30.2.3	<code>os</code>	143
6.31	PhotoFinish::jpeg_source_state_t Struct Reference	143
6.31.1	Detailed Description	144
6.31.2	Member Data Documentation	144
6.31.2.1	<code>buffer</code>	144
6.31.2.2	<code>buffer_size</code>	144
6.31.2.3	<code>is</code>	144
6.32	PhotoFinish::Kernel1Dvar Class Reference	145

6.32.1 Detailed Description . . . . .	146
6.32.2 Member Typedef Documentation . . . . .	146
6.32.2.1 ptr . . . . .	146
6.32.3 Constructor & Destructor Documentation . . . . .	146
6.32.3.1 Kernel1Dvar() [1/2] . . . . .	146
6.32.3.2 Kernel1Dvar() [2/2] . . . . .	146
6.32.3.3 ~Kernel1Dvar() . . . . .	147
6.32.4 Member Function Documentation . . . . .	147
6.32.4.1 build() . . . . .	147
6.32.4.2 convolve_h() . . . . .	147
6.32.4.3 convolve_h_type() . . . . .	148
6.32.4.4 convolve_h_type_channels() . . . . .	148
6.32.4.5 convolve_v() . . . . .	148
6.32.4.6 convolve_v_type() . . . . .	148
6.32.4.7 convolve_v_type_channels() . . . . .	149
6.32.4.8 create() . . . . .	149
6.32.4.9 eval() . . . . .	149
6.32.4.10 range() . . . . .	150
6.32.5 Member Data Documentation . . . . .	150
6.32.5.1 _scale . . . . .	150
6.32.5.2 _size . . . . .	150
6.32.5.3 _start . . . . .	150
6.32.5.4 _to_size . . . . .	150
6.32.5.5 _to_size_i . . . . .	150
6.32.5.6 _weights . . . . .	151
6.33 PhotoFinish::Kernel2D Class Reference . . . . .	151
6.33.1 Detailed Description . . . . .	152
6.33.2 Member Typedef Documentation . . . . .	152
6.33.2.1 ptr . . . . .	152
6.33.3 Constructor & Destructor Documentation . . . . .	152

6.33.3.1	<a href="#">Kernel2D()</a> [1/3]	153
6.33.3.2	<a href="#">Kernel2D()</a> [2/3]	153
6.33.3.3	<a href="#">Kernel2D()</a> [3/3]	153
6.33.3.4	<a href="#">~Kernel2D()</a>	153
6.33.4	<a href="#">Member Function Documentation</a>	153
6.33.4.1	<a href="#">convolve()</a>	153
6.33.4.2	<a href="#">convolve_type()</a>	154
6.33.4.3	<a href="#">convolve_type_channels()</a>	154
6.33.4.4	<a href="#">create()</a>	154
6.33.5	<a href="#">Member Data Documentation</a>	155
6.33.5.1	<a href="#">_centrex</a>	155
6.33.5.2	<a href="#">_centrey</a>	155
6.33.5.3	<a href="#">_height</a>	155
6.33.5.4	<a href="#">_values</a>	155
6.33.5.5	<a href="#">_width</a>	155
6.34	<a href="#">PhotoFinish::Lanczos Class Reference</a>	156
6.34.1	<a href="#">Detailed Description</a>	156
6.34.2	<a href="#">Constructor &amp; Destructor Documentation</a>	156
6.34.2.1	<a href="#">Lanczos()</a> [1/2]	156
6.34.2.2	<a href="#">Lanczos()</a> [2/2]	156
6.35	<a href="#">PhotoFinish::LibraryError Class Reference</a>	157
6.35.1	<a href="#">Detailed Description</a>	157
6.35.2	<a href="#">Constructor &amp; Destructor Documentation</a>	158
6.35.2.1	<a href="#">LibraryError()</a>	158
6.35.3	<a href="#">Member Function Documentation</a>	158
6.35.3.1	<a href="#">what()</a>	158
6.36	<a href="#">PhotoFinish::MemAllocError Class Reference</a>	158
6.36.1	<a href="#">Detailed Description</a>	159
6.36.2	<a href="#">Constructor &amp; Destructor Documentation</a>	159
6.36.2.1	<a href="#">MemAllocError()</a>	159



6.36.3	Member Function Documentation	159
6.36.3.1	what()	159
6.37	PhotoFinish::NoResults Class Reference	160
6.37.1	Detailed Description	160
6.37.2	Constructor & Destructor Documentation	160
6.37.2.1	NoResults()	160
6.37.3	Member Function Documentation	161
6.37.3.1	what()	161
6.37.4	Member Data Documentation	161
6.37.4.1	_class	161
6.37.4.2	_method	161
6.38	PhotoFinish::NoTargets Class Reference	161
6.38.1	Detailed Description	162
6.38.2	Constructor & Destructor Documentation	162
6.38.2.1	NoTargets()	162
6.38.3	Member Function Documentation	162
6.38.3.1	what()	162
6.38.4	Member Data Documentation	163
6.38.4.1	_destination	163
6.39	PhotoFinish::PNGreader_cb Struct Reference	163
6.39.1	Detailed Description	163
6.39.2	Constructor & Destructor Documentation	163
6.39.2.1	PNGreader_cb()	163
6.39.3	Member Function Documentation	164
6.39.3.1	end()	164
6.39.3.2	info()	164
6.39.3.3	row()	164
6.39.4	Member Data Documentation	164
6.39.4.1	_destination	164
6.39.4.2	_image	165

6.40 CMS::Profile Class Reference . . . . .	165
6.40.1 Detailed Description . . . . .	166
6.40.2 Member Typedef Documentation . . . . .	166
6.40.2.1 ptr . . . . .	167
6.40.3 Constructor & Destructor Documentation . . . . .	167
6.40.3.1 Profile() [1/5] . . . . .	167
6.40.3.2 Profile() [2/5] . . . . .	167
6.40.3.3 Profile() [3/5] . . . . .	167
6.40.3.4 Profile() [4/5] . . . . .	168
6.40.3.5 Profile() [5/5] . . . . .	168
6.40.3.6 ~Profile() . . . . .	168
6.40.4 Member Function Documentation . . . . .	168
6.40.4.1 copyright() . . . . .	168
6.40.4.2 copyright_wide() . . . . .	169
6.40.4.3 description() . . . . .	169
6.40.4.4 description_wide() . . . . .	169
6.40.4.5 Lab4() . . . . .	169
6.40.4.6 manufacturer() . . . . .	170
6.40.4.7 manufacturer_wide() . . . . .	170
6.40.4.8 model() . . . . .	170
6.40.4.9 model_wide() . . . . .	170
6.40.4.10 operator cmsHPROFILE() . . . . .	171
6.40.4.11 save_to_mem() . . . . .	171
6.40.4.12 set_copyright() [1/2] . . . . .	171
6.40.4.13 set_copyright() [2/2] . . . . .	171
6.40.4.14 set_description() [1/2] . . . . .	172
6.40.4.15 set_description() [2/2] . . . . .	172
6.40.4.16 set_manufacturer() [1/2] . . . . .	172
6.40.4.17 set_manufacturer() [2/2] . . . . .	172
6.40.4.18 set_model() [1/2] . . . . .	173

6.40.4.19	<a href="#">set_model()</a> [2/2]	173
6.40.4.20	<a href="#">sGrey()</a>	173
6.40.4.21	<a href="#">sRGB()</a>	173
6.40.5	<a href="#">Friends And Related Function Documentation</a>	173
6.40.5.1	<a href="#">__gnu_cxx::new_allocator&lt; Profile &gt;</a>	174
6.41	<a href="#">PhotoFinish::Role_Definable Class Reference</a>	174
6.41.1	<a href="#">Detailed Description</a>	175
6.41.2	<a href="#">Constructor &amp; Destructor Documentation</a>	175
6.41.2.1	<a href="#">Role_Definable()</a>	175
6.41.3	<a href="#">Member Function Documentation</a>	175
6.41.3.1	<a href="#">defined()</a>	175
6.41.3.2	<a href="#">set_defined()</a>	176
6.41.3.3	<a href="#">undefine()</a>	176
6.41.4	<a href="#">Friends And Related Function Documentation</a>	176
6.41.4.1	<a href="#">defined</a>	176
6.41.5	<a href="#">Member Data Documentation</a>	176
6.41.5.1	<a href="#">_defined</a>	176
6.42	<a href="#">PhotoFinish::SOLwriter Class Reference</a>	177
6.42.1	<a href="#">Detailed Description</a>	177
6.42.2	<a href="#">Constructor &amp; Destructor Documentation</a>	177
6.42.2.1	<a href="#">SOLwriter()</a>	177
6.42.3	<a href="#">Member Function Documentation</a>	177
6.42.3.1	<a href="#">preferred_format()</a>	178
6.42.3.2	<a href="#">write()</a>	178
6.43	<a href="#">PhotoFinish::Tags Class Reference</a>	178
6.43.1	<a href="#">Detailed Description</a>	179
6.43.2	<a href="#">Member Typedef Documentation</a>	179
6.43.2.1	<a href="#">ptr</a>	179
6.43.3	<a href="#">Constructor &amp; Destructor Documentation</a>	180
6.43.3.1	<a href="#">Tags()</a> [1/3]	180

6.43.3.2	<a href="#">Tags()</a> [2/3]	180
6.43.3.3	<a href="#">Tags()</a> [3/3]	180
6.43.4	<a href="#">Member Function Documentation</a>	180
6.43.4.1	<a href="#">add_resolution()</a>	180
6.43.4.2	<a href="#">add_searchpath()</a>	181
6.43.4.3	<a href="#">copy_from()</a>	181
6.43.4.4	<a href="#">copy_to()</a>	181
6.43.4.5	<a href="#">dupe()</a>	181
6.43.4.6	<a href="#">EXIFtags()</a>	181
6.43.4.7	<a href="#">IPTCtags()</a>	182
6.43.4.8	<a href="#">load()</a>	182
6.43.4.9	<a href="#">make_thumbnail()</a>	182
6.43.4.10	<a href="#">try_load()</a>	182
6.43.4.11	<a href="#">variables()</a>	183
6.43.4.12	<a href="#">XMPtags()</a>	183
6.44	<a href="#">PhotoFinish::Timer Class Reference</a>	183
6.44.1	<a href="#">Detailed Description</a>	183
6.44.2	<a href="#">Constructor &amp; Destructor Documentation</a>	184
6.44.2.1	<a href="#">Timer()</a>	184
6.44.3	<a href="#">Member Function Documentation</a>	184
6.44.3.1	<a href="#">elapsed()</a>	184
6.44.3.2	<a href="#">elapsed_ns()</a>	184
6.44.3.3	<a href="#">start()</a>	184
6.44.3.4	<a href="#">stop()</a>	185
6.45	<a href="#">CMS::Transform Class Reference</a>	185
6.45.1	<a href="#">Detailed Description</a>	186
6.45.2	<a href="#">Member Typedef Documentation</a>	186
6.45.2.1	<a href="#">ptr</a>	186
6.45.3	<a href="#">Constructor &amp; Destructor Documentation</a>	186
6.45.3.1	<a href="#">Transform()</a> [1/2]	186

6.45.3.2	<a href="#">Transform()</a> [2/2]	186
6.45.3.3	<a href="#">~Transform()</a>	187
6.45.4	<a href="#">Member Function Documentation</a>	187
6.45.4.1	<a href="#">change_formats()</a>	187
6.45.4.2	<a href="#">device_link()</a>	187
6.45.4.3	<a href="#">input_format()</a>	187
6.45.4.4	<a href="#">output_format()</a>	188
6.45.4.5	<a href="#">Proofing()</a>	188
6.45.4.6	<a href="#">transform_buffer()</a>	188
6.45.5	<a href="#">Friends And Related Function Documentation</a>	188
6.45.5.1	<a href="#">__gnu_cxx::new_allocator&lt; Transform &gt;</a>	188
6.46	<a href="#">PhotoFinish::Unimplemented Class Reference</a>	189
6.46.1	<a href="#">Detailed Description</a>	189
6.46.2	<a href="#">Constructor &amp; Destructor Documentation</a>	189
6.46.2.1	<a href="#">Unimplemented()</a>	189
6.46.3	<a href="#">Member Function Documentation</a>	190
6.46.3.1	<a href="#">what()</a>	190
6.46.4	<a href="#">Member Data Documentation</a>	190
6.46.4.1	<a href="#">_class</a>	190
6.46.4.2	<a href="#">_method</a>	190
6.47	<a href="#">PhotoFinish::Uninitialised Class Reference</a>	190
6.47.1	<a href="#">Detailed Description</a>	191
6.47.2	<a href="#">Constructor &amp; Destructor Documentation</a>	191
6.47.2.1	<a href="#">Uninitialised()</a> [1/2]	191
6.47.2.2	<a href="#">Uninitialised()</a> [2/2]	191
6.47.3	<a href="#">Member Function Documentation</a>	192
6.47.3.1	<a href="#">what()</a>	192
6.47.4	<a href="#">Member Data Documentation</a>	192
6.47.4.1	<a href="#">_attribute</a>	192
6.47.4.2	<a href="#">_class</a>	192

6.48 PhotoFinish::UnknownFileType Class Reference . . . . .	193
6.48.1 Detailed Description . . . . .	193
6.48.2 Constructor & Destructor Documentation . . . . .	193
6.48.2.1 UnknownFileType() [1/2] . . . . .	193
6.48.2.2 UnknownFileType() [2/2] . . . . .	194
6.48.3 Member Function Documentation . . . . .	194
6.48.3.1 what() . . . . .	194
6.49 PhotoFinish::webp_stream_writer Class Reference . . . . .	194
6.49.1 Detailed Description . . . . .	195
6.49.2 Constructor & Destructor Documentation . . . . .	195
6.49.2.1 webp_stream_writer() . . . . .	195
6.49.2.2 ~webp_stream_writer() . . . . .	196
6.49.3 Member Function Documentation . . . . .	196
6.49.3.1 add_exif() . . . . .	196
6.49.3.2 add_icc() . . . . .	196
6.49.3.3 add_xmp() . . . . .	196
6.49.3.4 after_chunk() . . . . .	197
6.49.3.5 before_chunk() . . . . .	197
6.49.3.6 modify_chunk() . . . . .	197
6.49.3.7 modify_vp8x() . . . . .	197
6.49.3.8 write() . . . . .	197
6.49.3.9 write_chunk() . . . . .	198
6.50 PhotoFinish::WebPError Class Reference . . . . .	198
6.50.1 Detailed Description . . . . .	198
6.50.2 Constructor & Destructor Documentation . . . . .	198
6.50.2.1 WebPError() . . . . .	198
6.50.3 Member Function Documentation . . . . .	199
6.50.3.1 what() . . . . .	199

<b>7 File Documentation</b>	<b>201</b>
7.1 Benchmark.cc File Reference	201
7.2 Benchmark.hh File Reference	201
7.3 CMS.cc File Reference	202
7.3.1 Macro Definition Documentation	203
7.3.1.1 BYTES_MASK	203
7.3.1.2 CHANNELS_MASK	203
7.3.1.3 COLORSPACE_MASK	203
7.3.1.4 DOSWAP_MASK	203
7.3.1.5 ENDIAN16_MASK	203
7.3.1.6 EXTRA_MASK	204
7.3.1.7 FLAVOR_MASK	204
7.3.1.8 FLOAT_MASK	204
7.3.1.9 OPTIMIZED_MASK	204
7.3.1.10 PLANAR_MASK	204
7.3.1.11 SWAPFIRST_MASK	204
7.3.2 Function Documentation	205
7.3.2.1 lcms2_error_adaptor()	205
7.3.2.2 lcms2_errorhandler()	205
7.4 CMS.hh File Reference	205
7.4.1 Function Documentation	206
7.4.1.1 lcms2_error_adaptor()	207
7.5 CropSolution.cc File Reference	207
7.5.1 Macro Definition Documentation	207
7.5.1.1 max	207
7.5.1.2 min	208
7.5.1.3 sqr	208
7.6 CropSolution.hh File Reference	208
7.7 Definable.hh File Reference	209
7.8 Destination.cc File Reference	209

7.9	Destination.hh File Reference . . . . .	209
7.10	Destination_items.cc File Reference . . . . .	210
7.11	Destination_items.hh File Reference . . . . .	210
7.12	Ditherer.cc File Reference . . . . .	211
7.12.1	Macro Definition Documentation . . . . .	212
7.12.1.1	nextpos . . . . .	212
7.12.1.2	pos . . . . .	212
7.12.1.3	prevpos . . . . .	212
7.13	Ditherer.hh File Reference . . . . .	212
7.14	Exception.hh File Reference . . . . .	213
7.15	Frame.cc File Reference . . . . .	214
7.16	Frame.hh File Reference . . . . .	214
7.17	Image.cc File Reference . . . . .	214
7.18	Image.hh File Reference . . . . .	215
7.19	ImageFile.cc File Reference . . . . .	216
7.20	ImageFile.hh File Reference . . . . .	216
7.21	JP2.hh File Reference . . . . .	217
7.22	JP2_callbacks.cc File Reference . . . . .	217
7.23	JP2reader.cc File Reference . . . . .	218
7.24	JP2writer.cc File Reference . . . . .	218
7.25	JPEG.hh File Reference . . . . .	218
7.26	JPEG_iostream.cc File Reference . . . . .	219
7.27	JPEG_profiles.cc File Reference . . . . .	220
7.28	JPEG_scans.cc File Reference . . . . .	221
7.29	JPEGreader.cc File Reference . . . . .	221
7.30	JPEGwriter.cc File Reference . . . . .	221
7.31	JXR.hh File Reference . . . . .	222
7.31.1	Macro Definition Documentation . . . . .	222
7.31.1.1	FmtPair . . . . .	222
7.31.1.2	JXRcheck . . . . .	223



7.32 JXR_formats.cc File Reference . . . . .	223
7.33 JXRreader.cc File Reference . . . . .	223
7.33.1 Macro Definition Documentation . . . . .	224
7.33.1.1 jxr_metadata_data . . . . .	224
7.33.1.2 jxr_metadata_size . . . . .	224
7.34 JXRwriter.cc File Reference . . . . .	224
7.35 Kernel1Dvar.cc File Reference . . . . .	225
7.35.1 Macro Definition Documentation . . . . .	225
7.35.1.1 min . . . . .	225
7.35.1.2 sqr . . . . .	225
7.36 Kernel1Dvar.hh File Reference . . . . .	225
7.37 Kernel2D.cc File Reference . . . . .	226
7.37.1 Macro Definition Documentation . . . . .	226
7.37.1.1 sqr . . . . .	226
7.38 Kernel2D.hh File Reference . . . . .	226
7.39 LCMS2ErrorHandler.cc File Reference . . . . .	227
7.40 photofinish.cc File Reference . . . . .	227
7.40.1 Function Documentation . . . . .	228
7.40.1.1 main() . . . . .	228
7.41 PNGreader.cc File Reference . . . . .	228
7.42 PNGreader_cb.cc File Reference . . . . .	228
7.43 PNGreader_cb.hh File Reference . . . . .	229
7.44 PNGwriter.cc File Reference . . . . .	229
7.45 process_scans.cc File Reference . . . . .	230
7.45.1 Function Documentation . . . . .	230
7.45.1.1 main() . . . . .	231
7.45.1.2 make_preview() . . . . .	231
7.45.1.3 preview_dir() . . . . .	231
7.46 sample.h File Reference . . . . .	231
7.46.1 Macro Definition Documentation . . . . .	231

7.46.1.1	SAMPLE	231
7.46.1.2	SET_SAMPLE_FORMAT	232
7.47	SOLwriter.cc File Reference	232
7.48	Tags.cc File Reference	232
7.49	Tags.hh File Reference	233
7.49.1	Macro Definition Documentation	234
7.49.1.1	StrPair	234
7.50	Tags_EXIF_subst.cc File Reference	234
7.50.1	Macro Definition Documentation	234
7.50.1.1	Key	235
7.51	Tags_IPTC_subst.cc File Reference	235
7.52	Tags_XMP_subst.cc File Reference	235
7.53	TIFFreader.cc File Reference	236
7.53.1	Macro Definition Documentation	236
7.53.1.1	TIFFcheck	236
7.54	TIFFwriter.cc File Reference	236
7.54.1	Macro Definition Documentation	237
7.54.1.1	TIFFcheck	237
7.55	WebP_ostream.cc File Reference	237
7.55.1	Macro Definition Documentation	237
7.55.1.1	min	238
7.56	WebP_ostream.hh File Reference	238
7.57	WebPreader.cc File Reference	238
7.58	WebPwriter.cc File Reference	239
<b>Index</b>		<b>241</b>

# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">CMS</a> . . . . .	<a href="#">11</a>
<a href="#">PhotoFinish</a> . . . . .	<a href="#">16</a>



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

PhotoFinish::CropSolver . . . . .	40
PhotoFinish::D_profile . . . . .	54
PhotoFinish::D_target . . . . .	62
PhotoFinish::Frame . . . . .	118
PhotoFinish::definable< T > . . . . .	74
PhotoFinish::definable< bool > . . . . .	74
PhotoFinish::definable< CMS::Intent > . . . . .	74
PhotoFinish::definable< double > . . . . .	74
PhotoFinish::definable< fs::path > . . . . .	74
PhotoFinish::definable< int > . . . . .	74
PhotoFinish::definable< std::pair< int, int > > . . . . .	74
PhotoFinish::definable< std::string > . . . . .	74
PhotoFinish::definable< unsigned char > . . . . .	74
PhotoFinish::Destination . . . . .	78
PhotoFinish::Destinations . . . . .	89
PhotoFinish::Ditherer . . . . .	93
exception	
PhotoFinish::ErrorMsg . . . . .	95
PhotoFinish::cmsTypeError . . . . .	39
PhotoFinish::DestinationError . . . . .	88
PhotoFinish::FileError . . . . .	98
PhotoFinish::FileContentError . . . . .	97
PhotoFinish::FileOpenError . . . . .	101
PhotoFinish::UnknownFileType . . . . .	193
PhotoFinish::LibraryError . . . . .	157
PhotoFinish::MemAllocError . . . . .	158
PhotoFinish::NoResults . . . . .	160
PhotoFinish::NoTargets . . . . .	161
PhotoFinish::Unimplemented . . . . .	189
PhotoFinish::Uninitialised . . . . .	190
PhotoFinish::WebPError . . . . .	198
CMS::Format . . . . .	102
PhotoFinish::Image . . . . .	123
PhotoFinish::ImageFilepath . . . . .	133
PhotoFinish::ImageReader . . . . .	136

PhotoFinish::ImageWriter . . . . .	139
PhotoFinish::SOLwriter . . . . .	177
PhotoFinish::jpeg_destination_state_t . . . . .	142
PhotoFinish::jpeg_source_state_t . . . . .	143
PhotoFinish::Kernel1Dvar . . . . .	145
PhotoFinish::Lanczos . . . . .	156
PhotoFinish::Kernel2D . . . . .	151
PhotoFinish::GaussianSharpen . . . . .	122
PhotoFinish::PNGreader_cb . . . . .	163
CMS::Profile . . . . .	165
PhotoFinish::Role_Definable . . . . .	174
PhotoFinish::D_JP2 . . . . .	41
PhotoFinish::D_JPEG . . . . .	46
PhotoFinish::D_JXR . . . . .	49
PhotoFinish::D_PNG . . . . .	53
PhotoFinish::D_resize . . . . .	59
PhotoFinish::D_sharpen . . . . .	61
PhotoFinish::D_thumbnail . . . . .	66
PhotoFinish::D_TIFF . . . . .	67
PhotoFinish::D_WebP . . . . .	70
PhotoFinish::Tags . . . . .	178
PhotoFinish::Timer . . . . .	183
CMS::Transform . . . . .	185
PhotoFinish::webp_stream_writer . . . . .	194

## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">PhotoFinish::cmsTypeError</a>	39
<a href="#">PhotoFinish::CropSolver</a>	
Class for finding the best frame position for cropping	40
<a href="#">PhotoFinish::D_JP2</a>	
JP2 parameters for destination	41
<a href="#">PhotoFinish::D_JPEG</a>	
JPEG parameters for destination	46
<a href="#">PhotoFinish::D_JXR</a>	
JPEG XR parameters for destination	49
<a href="#">PhotoFinish::D_PNG</a>	
PNG parameters for destination	53
<a href="#">PhotoFinish::D_profile</a>	
ICC profile parameters for destination	54
<a href="#">PhotoFinish::D_resize</a>	
Resize parameters for destination	59
<a href="#">PhotoFinish::D_sharpen</a>	
Sharpen parameters for destination	61
<a href="#">PhotoFinish::D_target</a>	
Target parameters for destination	62
<a href="#">PhotoFinish::D_thumbnail</a>	
Thumbnail parameters for destination	66
<a href="#">PhotoFinish::D_TIFF</a>	
TIFF parameters for destination	67
<a href="#">PhotoFinish::D_WebP</a>	
WebP parameters for destination	70
<a href="#">PhotoFinish::definable&lt; T &gt;</a>	
Template class for storing things that can be defined or undefined	74
<a href="#">PhotoFinish::Destination</a>	
Represents a destination, read from destinations.yml	78
<a href="#">PhotoFinish::DestinationError</a>	
Destination exception	88
<a href="#">PhotoFinish::Destinations</a>	
A wrapper class for reading destinations from a YAML file and storing them in a map	89
<a href="#">PhotoFinish::Ditherer</a>	
Class for dithering images down to 8-bit components	93

<a href="#">PhotoFinish::ErrorMsg</a>	
Generic error message exception . . . . .	95
<a href="#">PhotoFinish::FileContentError</a>	
File content exception . . . . .	97
<a href="#">PhotoFinish::FileError</a>	
File error abstract base exception . . . . .	98
<a href="#">PhotoFinish::FileOpenError</a>	
File open exception . . . . .	101
<a href="#">CMS::Format</a>	
Wrap LCMS2's pixel format . . . . .	102
<a href="#">PhotoFinish::Frame</a>	
Crop+rescaling parameters . . . . .	118
<a href="#">PhotoFinish::GaussianSharpen</a>	
GaussianSharpen kernel . . . . .	122
<a href="#">PhotoFinish::Image</a>	
An image class . . . . .	123
<a href="#">PhotoFinish::ImageFilepath</a>	
Class for holding filename and the image format . . . . .	133
<a href="#">PhotoFinish::ImageReader</a>	
Abstract base class for reading image files . . . . .	136
<a href="#">PhotoFinish::ImageWriter</a>	
Abstract base class for writing image files . . . . .	139
<a href="#">PhotoFinish::jpeg_destination_state_t</a>	
Structure holding information for the ostream writer . . . . .	142
<a href="#">PhotoFinish::jpeg_source_state_t</a>	
Structure holding information for the istream reader . . . . .	143
<a href="#">PhotoFinish::Kernel1Dvar</a>	
Creates and stores coefficients for cropping and resizing an image . . . . .	145
<a href="#">PhotoFinish::Kernel2D</a>	
Creates and stores coefficients for convolving an image . . . . .	151
<a href="#">PhotoFinish::Lanczos</a>	
Lanczos filter . . . . .	156
<a href="#">PhotoFinish::LibraryError</a>	
Library exception . . . . .	157
<a href="#">PhotoFinish::MemAllocError</a>	
Memory allocation exception . . . . .	158
<a href="#">PhotoFinish::NoResults</a>	
No results exception . . . . .	160
<a href="#">PhotoFinish::NoTargets</a>	
No targets exception . . . . .	161
<a href="#">PhotoFinish::PNGreader_cb</a>	
	163
<a href="#">CMS::Profile</a>	
Wrap LCMS2's cmsHPROFILE . . . . .	165
<a href="#">PhotoFinish::Role_Definable</a>	
Base class for adding "definable" attribute . . . . .	174
<a href="#">PhotoFinish::SOLwriter</a>	
Write the boot logo files for use on Motorola Atrix 4G and possibly other phones . . . . .	177
<a href="#">PhotoFinish::Tags</a>	
Reads and holds tag information . . . . .	178
<a href="#">PhotoFinish::Timer</a>	
Class for doing nanosecond-accurate timings . . . . .	183
<a href="#">CMS::Transform</a>	
Wrap LCMS2's transform object . . . . .	185
<a href="#">PhotoFinish::Unimplemented</a>	
Unimplemented method exception . . . . .	189
<a href="#">PhotoFinish::Uninitialised</a>	
Uninitialised attribute exception . . . . .	190



---

<a href="#">PhotoFinish::UnknownFileType</a>	
Unknown file type exception . . . . .	193
<a href="#">PhotoFinish::webp_stream_writer</a>	
A custom writer for libwebp that writes using a std::ostream object . . . . .	194
<a href="#">PhotoFinish::WebPError</a>	
WebP exception . . . . .	198



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

Benchmark.cc	201
Benchmark.hh	201
CMS.cc	202
CMS.hh	205
CropSolution.cc	207
CropSolution.hh	208
Definable.hh	209
Destination.cc	209
Destination.hh	209
Destination_items.cc	210
Destination_items.hh	210
Ditherer.cc	211
Ditherer.hh	212
Exception.hh	213
Frame.cc	214
Frame.hh	214
Image.cc	214
Image.hh	215
ImageFile.cc	216
ImageFile.hh	216
JP2.hh	217
JP2_callbacks.cc	217
JP2reader.cc	218
JP2writer.cc	218
JPEG.hh	218
JPEG_iostream.cc	219
JPEG_profiles.cc	220
JPEG_scans.cc	221
JPEGreader.cc	221
JPEGwriter.cc	221
JXR.hh	222
JXR_formats.cc	223
JXRreader.cc	223
JXRwriter.cc	224
Kernel1Dvar.cc	225

Kernel1Dvar.hh	225
Kernel2D.cc	226
Kernel2D.hh	226
LCMS2ErrorHandler.cc	227
photofinish.cc	227
PNGreader.cc	228
PNGreader_cb.cc	228
PNGreader_cb.hh	229
PNGwriter.cc	229
process_scans.cc	230
sample.h	231
SOLwriter.cc	232
Tags.cc	232
Tags.hh	233
Tags_EXIF_subst.cc	234
Tags_IPTC_subst.cc	235
Tags_XMP_subst.cc	235
TIFFreader.cc	236
TIFFwriter.cc	236
WebP_ostream.cc	237
WebP_ostream.hh	238
WebPreader.cc	238
WebPwriter.cc	239

## Chapter 5

# Namespace Documentation

### 5.1 CMS Namespace Reference

#### Classes

- class [Format](#)  
*Wrap LCMS2's pixel format.*
- class [Profile](#)  
*Wrap LCMS2's cmsHPROFILE.*
- class [Transform](#)  
*Wrap LCMS2's transform object.*

#### Enumerations

- enum [ColourModel](#) {  
[ColourModel::Any](#) = 0, [ColourModel::Greyscale](#) = 3, [ColourModel::RGB](#), [ColourModel::CMY](#),  
[ColourModel::CMYK](#), [ColourModel::YCbCr](#), [ColourModel::YUV](#), [ColourModel::XYZ](#),  
[ColourModel::Lab](#), [ColourModel::YUVK](#), [ColourModel::HSV](#), [ColourModel::HLS](#),  
[ColourModel::Yxy](#), [ColourModel::MCH1](#), [ColourModel::MCH2](#), [ColourModel::MCH3](#),  
[ColourModel::MCH4](#), [ColourModel::MCH5](#), [ColourModel::MCH6](#), [ColourModel::MCH7](#),  
[ColourModel::MCH8](#), [ColourModel::MCH9](#), [ColourModel::MCH10](#), [ColourModel::MCH11](#),  
[ColourModel::MCH12](#), [ColourModel::MCH13](#), [ColourModel::MCH14](#), [ColourModel::MCH15](#),  
[ColourModel::LabV2](#) }  
*An enum class of LCMS2's colour models.*
- enum [Intent](#) {  
[Intent::Perceptual](#), [Intent::Relative\\_colormetric](#), [Intent::Saturation](#), [Intent::Absolute\\_colormetric](#),  
[Intent::Preserve\\_k\\_only\\_perceptual](#) = 10, [Intent::Preserve\\_k\\_only\\_relative\\_colormetric](#), [Intent::Preserve\\_k\\_↵](#)  
[k\\_only\\_saturation](#), [Intent::Preserve\\_k\\_only\\_absolute\\_colormetric](#),  
[Intent::Preserve\\_k\\_plane\\_perceptual](#), [Intent::Preserve\\_k\\_plane\\_relative\\_colormetric](#), [Intent::Preserve\\_k\\_↵](#)  
[plane\\_saturation](#), [Intent::Preserve\\_k\\_plane\\_absolute\\_colormetric](#) }  
*Wrap LCMS2's intents.*

## Functions

- `std::ostream & operator<<` (`std::ostream &out`, [ColourModel](#) model)
- `std::ostream & operator<<` (`std::ostream &out`, [Format](#) f)
- `cmsIOHANDLER * OpenIOHandlerFromIStream` (`std::istream *is`)
- `cmsIOHANDLER * OpenIOHandlerFromIFStream` (`fs::path filepath`)
- `cmsUInt32Number istream_read` (`cmsIOHANDLER *iohandler`, `void *Buffer`, `cmsUInt32Number size`, `cmsUInt32Number count`)
- `cmsBool istream_seek` (`cmsIOHANDLER *iohandler`, `cmsUInt32Number offset`)
- `cmsBool istream_close` (`cmsIOHANDLER *iohandler`)
- `cmsUInt32Number istream_tell` (`cmsIOHANDLER *iohandler`)
- `cmsBool istream_write` (`cmsIOHANDLER *iohandler`, `cmsUInt32Number size`, `const void *Buffer`)
- `cmsUInt32Number ostream_read` (`cmsIOHANDLER *iohandler`, `void *Buffer`, `cmsUInt32Number size`, `cmsUInt32Number count`)
- `cmsBool ostream_seek` (`cmsIOHANDLER *iohandler`, `cmsUInt32Number offset`)
- `cmsBool ostream_close` (`cmsIOHANDLER *iohandler`)
- `cmsUInt32Number ostream_tell` (`cmsIOHANDLER *iohandler`)
- `cmsBool ostream_write` (`cmsIOHANDLER *iohandler`, `cmsUInt32Number size`, `const void *Buffer`)

## 5.1.1 Enumeration Type Documentation

### 5.1.1.1 ColourModel

```
enum CMS::ColourModel [strong]
```

An enum class of LCMS2's colour models.

#### Enumerator

Any	
Greyscale	
RGB	
CMY	
CMYK	
YCbCr	
YUV	
XYZ	
Lab	
YUVK	
HSV	
HLS	
Yxy	
MCH1	
MCH2	
MCH3	
MCH4	
MCH5	
MCH6	
MCH7	
MCH8	

## Enumerator

MCH9	
MCH10	
MCH11	
MCH12	
MCH13	
MCH14	
MCH15	
LabV2	

Definition at line 137 of file CMS.hh.

## 5.1.1.2 Intent

```
enum CMS::Intent [strong]
```

Wrap LCMS2's intents.

## Enumerator

Perceptual	
Relative_colormetric	
Saturation	
Absolute_colormetric	
Preserve_k_only_perceptual	
Preserve_k_only_relative_colormetric	
Preserve_k_only_saturation	
Preserve_k_only_absolute_colormetric	
Preserve_k_plane_perceptual	
Preserve_k_plane_relative_colormetric	
Preserve_k_plane_saturation	
Preserve_k_plane_absolute_colormetric	

Definition at line 378 of file CMS.hh.

## 5.1.2 Function Documentation

## 5.1.2.1 istream\_close()

```
cmsBool CMS::istream_close (
    cmsIOHANDLER * iohandler )
```

Definition at line 646 of file CMS.cc.

#### 5.1.2.2 `istream_read()`

```
cmsUInt32Number CMS::istream_read (
    cmsIOHANDLER * iohandler,
    void * Buffer,
    cmsUInt32Number size,
    cmsUInt32Number count )
```

Definition at line 632 of file CMS.cc.

#### 5.1.2.3 `istream_seek()`

```
cmsBool CMS::istream_seek (
    cmsIOHANDLER * iohandler,
    cmsUInt32Number offset )
```

Definition at line 639 of file CMS.cc.

#### 5.1.2.4 `istream_tell()`

```
cmsUInt32Number CMS::istream_tell (
    cmsIOHANDLER * iohandler )
```

Definition at line 658 of file CMS.cc.

#### 5.1.2.5 `istream_write()`

```
cmsBool CMS::istream_write (
    cmsIOHANDLER * iohandler,
    cmsUInt32Number size,
    const void * Buffer )
```

Definition at line 663 of file CMS.cc.

#### 5.1.2.6 `OpenIOhandlerFromIFStream()`

```
cmsIOHANDLER * CMS::OpenIOhandlerFromIFStream (
    fs::path filepath )
```

Definition at line 623 of file CMS.cc.



#### 5.1.2.7 OpenIOhandlerFromIStream()

```
cmsIOHANDLER * CMS::OpenIOhandlerFromIStream (
    std::istream * is )
```

Definition at line 601 of file CMS.cc.

#### 5.1.2.8 operator<<() [1/2]

```
std::ostream & CMS::operator<< (
    std::ostream & out,
    ColourModel model )
```

Definition at line 228 of file CMS.cc.

#### 5.1.2.9 operator<<() [2/2]

```
std::ostream & CMS::operator<< (
    std::ostream & out,
    Format f )
```

Definition at line 497 of file CMS.cc.

#### 5.1.2.10 ostream\_close()

```
cmsBool CMS::ostream_close (
    cmsIOHANDLER * iohandler )
```

Definition at line 680 of file CMS.cc.

#### 5.1.2.11 ostream\_read()

```
cmsUInt32Number CMS::ostream_read (
    cmsIOHANDLER * iohandler,
    void * Buffer,
    cmsUInt32Number size,
    cmsUInt32Number count )
```

Definition at line 668 of file CMS.cc.

#### 5.1.2.12 ostream\_seek()

```
cmsBool CMS::ostream_seek (
    cmsIOHANDLER * iohandler,
    cmsUInt32Number offset )
```

Definition at line 673 of file CMS.cc.

#### 5.1.2.13 ostream\_tell()

```
cmsUInt32Number CMS::ostream_tell (
    cmsIOHANDLER * iohandler )
```

Definition at line 692 of file CMS.cc.

#### 5.1.2.14 ostream\_write()

```
cmsBool CMS::ostream_write (
    cmsIOHANDLER * iohandler,
    cmsUInt32Number size,
    const void * Buffer )
```

Definition at line 697 of file CMS.cc.

## 5.2 PhotoFinish Namespace Reference

### Classes

- class [cmsTypeError](#)
- class [CropSolver](#)  
*Class for finding the best frame position for cropping.*
- class [D\\_JP2](#)  
*JP2 parameters for destination.*
- class [D\\_JPEG](#)  
*JPEG parameters for destination.*
- class [D\\_JXR](#)  
*JPEG XR parameters for destination.*
- class [D\\_PNG](#)  
*PNG parameters for destination.*
- class [D\\_profile](#)  
*ICC profile parameters for destination.*
- class [D\\_resize](#)  
*Resize parameters for destination.*
- class [D\\_sharpen](#)  
*Sharpen parameters for destination.*

- class [D\\_target](#)  
*Target parameters for destination.*
- class [D\\_thumbnail](#)  
*Thumbnail parameters for destination.*
- class [D\\_TIFF](#)  
*TIFF parameters for destination.*
- class [D\\_WebP](#)  
*WebP parameters for destination.*
- class [definable](#)  
*Template class for storing things that can be defined or undefined.*
- class [Destination](#)  
*Represents a destination, read from destinations.yml.*
- class [DestinationError](#)  
*[Destination](#) exception.*
- class [Destinations](#)  
*A wrapper class for reading destinations from a YAML file and storing them in a map.*
- class [Ditherer](#)  
*Class for dithering images down to 8-bit components.*
- class [ErrorMsg](#)  
*Generic error message exception.*
- class [FileContentError](#)  
*File content exception.*
- class [FileError](#)  
*File error abstract base exception.*
- class [FileOpenError](#)  
*File open exception.*
- class [Frame](#)  
*Crop+rescaling parameters.*
- class [GaussianSharpen](#)  
*[GaussianSharpen](#) kernel.*
- class [Image](#)  
*An image class.*
- class [ImageFilepath](#)  
*Class for holding filename and the image format.*
- class [ImageReader](#)  
*Abstract base class for reading image files.*
- class [ImageWriter](#)  
*Abstract base class for writing image files.*
- struct [jpeg\\_destination\\_state\\_t](#)  
*Structure holding information for the ostream writer.*
- struct [jpeg\\_source\\_state\\_t](#)  
*Structure holding information for the istream reader.*
- class [Kernel1Dvar](#)  
*Creates and stores coefficients for cropping and resizing an image.*
- class [Kernel2D](#)  
*Creates and stores coefficients for convolving an image.*
- class [Lanczos](#)  
*[Lanczos](#) filter.*
- class [LibraryError](#)  
*Library exception.*
- class [MemAllocError](#)

- Memory allocation exception.*
- class [NoResults](#)
  - No results exception.*
- class [NoTargets](#)
  - No targets exception.*
- struct [PNGreader\\_cb](#)
- class [Role\\_Definable](#)
  - Base class for adding "definable" attribute.*
- class [SOLwriter](#)
  - Write the boot logo files for use on Motorola Atrix 4G and possibly other phones.*
- class [Tags](#)
  - Reads and holds tag information.*
- class [Timer](#)
  - Class for doing nanosecond-accurate timings.*
- class [Unimplemented](#)
  - [Unimplemented](#) method exception.*
- class [Uninitialised](#)
  - [Uninitialised](#) attribute exception.*
- class [UnknownFileType](#)
  - Unknown file type exception.*
- class [webp\\_stream\\_writer](#)
  - A custom writer for libwebp that writes using a std::ostream object.*
- class [WebPError](#)
  - WebP exception.*

## Typedefs

- typedef std::pair< double, double > [rulerpair](#)
  - Ruler paramaters - percentage of final image vs. pixel position in original.*
- typedef std::list< [rulerpair](#) > [rulerlist](#)
  - A list of rulers.*
- typedef std::map< std::string, std::string > [hash](#)
  - A simple hash.*
- typedef std::vector< std::string > [stringlist](#)
  - A list of strings.*
- typedef std::map< std::string, [stringlist](#) > [multihash](#)
  - A hash of string lists.*
- typedef std::vector< std::pair< std::string, std::string > > [subst\\_table](#)
- typedef std::vector< std::pair< unsigned int, const PKPixelFormatGUID \* > > [jxr\\_format\\_subst](#)

## Functions

- std::ostream & [operator<<](#) (std::ostream &out, [Timer](#) t)
- template<typename T >  
T [scaleval](#) (void)
  - A template function that returns the 'scale' value of a type.*
- template<>  
unsigned char [scaleval](#)< unsigned char > (void)
- template<>  
unsigned short int [scaleval](#)< unsigned short int > (void)

- `template<>`  
`unsigned int scaleval< unsigned int > (void)`
- `template<>`  
`unsigned long long scaleval< unsigned long long > (void)`
- `template<>`  
`float scaleval< float > (void)`
- `template<>`  
`double scaleval< double > (void)`
- `template<typename T >`  
`T limitval (SAMPLE v)`  
*A template function that limits a floating-point value while converting to another type.*
- `template<>`  
`unsigned char limitval< unsigned char > (SAMPLE v)`
- `template<>`  
`unsigned short int limitval< unsigned short int > (SAMPLE v)`
- `template<>`  
`unsigned int limitval< unsigned int > (SAMPLE v)`
- `template<>`  
`unsigned long long limitval< unsigned long long > (SAMPLE v)`
- `template<>`  
`float limitval< float > (SAMPLE v)`
- `template<>`  
`double limitval< double > (SAMPLE v)`
- `bool exists (const ImageFilepath &fp)`
- `std::time_t last\_write\_time (const ImageFilepath &fp)`
- `template<typename Num_type , typename R_type >`  
`Exiv2::ValueType< R_type > & closest\_Rational (double value)`  
*Find a close rational fraction given a floating-point value.*
- `void add\_rulers (multihash &vars, std::string key, rulerlist &rulers)`  
*Parse named variables into a list of rulers.*
- `void add\_ruler\_pins (rulerlist &rulers, unsigned int max)`  
*Add rulers to the either side of an image if there aren't enough.*
- `void error\_callback (const char *msg, void *client_data)`  
*Error callback for OpenJPEG - throw a [LibraryError](#) exception.*
- `void warning\_callback (const char *msg, void *client_data)`  
*Warning callback for OpenJPEG - print the message to STDERR.*
- `void info\_callback (const char *msg, void *client_data)`  
*Info callback for OpenJPEG - print the indented message to STDERR.*
- `template<typename T >`  
`void read\_planar (unsigned int width, unsigned char channels, opj\_image\_t *image, T *row, unsigned int y)`  
*Read a row of image data from OpenJPEG's planar integer components into an LCMS2-compatible single array.*
- `template<typename T >`  
`void write\_planar (unsigned int width, unsigned char channels, T *row, opj\_image\_t *image, unsigned int y)`  
*Read a row of planar pixel data into OpenJPEG's planar components.*
- `template<typename T >`  
`void write\_packed (unsigned int width, unsigned char channels, T *row, opj\_image\_t *image, unsigned int y)`  
*Read a row of packed pixel data into OpenJPEG's planar components.*
- `void jpeg\_istream\_src (j\_decompress\_ptr dinfo, std::istream *is)`  
*Set up a "source manager" on the given JPEG decompression structure to read from an istream.*
- `void jpeg\_istream\_src\_free (j\_decompress\_ptr dinfo)`  
*Free the data structures of the istream source manager.*
- `void jpeg\_ostream\_dest (j\_compress\_ptr cinfo, std::ostream *os)`  
*Setup a "destination manager" on the given JPEG compression structure to write to an ostream.*

- void [jpeg\\_ostream\\_dest\\_free](#) (j\_compress\_ptr cinfo)  
*Free the data structures of the ostream destination manager.*
- void [jpegfile\\_scan\\_RGB](#) (jpeg\_compress\_struct \*cinfo)  
*Create a scan "script" for an RGB image.*
- void [jpegfile\\_scan\\_grayscale](#) (jpeg\_compress\_struct \*cinfo)  
*Create a scan "script" for a grayscale image.*
- [CMS::Profile::ptr jpeg\\_read\\_profile](#) (jpeg\_decompress\_struct \*dinfo, [Destination::ptr](#) dest)  
*Read an ICC profile from APP2 markers in a JPEG file.*
- void [jpeg\\_write\\_profile](#) (jpeg\_compress\_struct \*cinfo, unsigned char \*data, unsigned int size)  
*Write an ICC profile into APP2 markers in a JPEG file.*
- void [jpeg\\_istream\\_init\\_source](#) (j\_decompress\_ptr dinfo)  
*Initialise the istream source manager.*
- boolean [jpeg\\_istream\\_fill\\_input\\_buffer](#) (j\_decompress\_ptr dinfo)  
*Fill the buffer.*
- void [jpeg\\_istream\\_skip\\_input\\_data](#) (j\_decompress\_ptr dinfo, long num\_bytes)  
*Skip some data.*
- boolean [jpeg\\_istream\\_resync\\_to\\_restart](#) (j\_decompress\_ptr dinfo, int desired)  
*Resync to start?!?*
- void [jpeg\\_istream\\_term\\_source](#) (j\_decompress\_ptr dinfo)  
*Terminate the istream source manager.*
- void [jpeg\\_error\\_exit](#) (j\_common\_ptr cinfo)
- const PKPixelFormatGUID & [jxr\\_pixel\\_format](#) (unsigned int n)
- [CMS::Format](#) [jxr\\_cms\\_format](#) (const PKPixelFormatGUID &g)
- void [png\\_info\\_cb](#) (png\_structp png, png\_infop info)  
*Called by libPNG when the iHDR chunk has been read with the main "header" information.*
- void [png\\_row\\_cb](#) (png\_structp png, png\_bytep row\_data, png\_uint\_32 row\_num, int pass)  
*Called by libPNG when a row of image data has been read.*
- void [png\\_end\\_cb](#) (png\_structp png, png\_infop info)  
*Called by libPNG when the image data has finished.*
- void [png\\_write\\_ostream\\_cb](#) (png\_structp png, png\_bytep buffer, png\_size\_t length)  
*libPNG callback for writing to an ostream*
- void [png\\_flush\\_ostream\\_cb](#) (png\_structp png)  
*libPNG callback for flushing an ostream*
- void [write\\_be](#) (void \*ptr, size\_t size, std::ostream &stream)
- int [webp\\_stream\\_writer\\_func](#) (const uint8\_t \*data, size\_t data\_size, const WebPPicture \*picture)  
*Wrapper around the [webp\\_stream\\_writer](#) class.*
- void [copy\\_le\\_to](#) (unsigned char \*dest, unsigned int value, unsigned char length)
- unsigned int [read\\_le32](#) (const unsigned char \*data)
- template<typename A, typename B >  
void [transfer\\_alpha\\_typed2](#) (unsigned int width, unsigned char src\_channels, const A \*src\_row, unsigned char dest\_channels, const B \*dest\_row)
- template<typename A >  
void [transfer\\_alpha\\_typed](#) (unsigned int width, unsigned char src\_channels, const A \*src\_row, [CMS::Format](#) dest\_format, const unsigned char \*dest\_row)
- void [transfer\\_alpha](#) (unsigned int width, [CMS::Format](#) src\_format, const unsigned char \*src\_row, [CMS::Format](#) dest\_format, const unsigned char \*dest\_row)
- std::string [profile\\_name](#) ([CMS::Profile::ptr](#) profile)
- void [lcms2\\_errorhandler](#) (cmsContext ContextID, cmsUInt32Number ErrorCode, const char \*Text)  
*Throw a [LibraryError](#) exception when LCMS2 returns an error.*
- void [lcms2\\_error\\_adaptor](#) (void)  
*Set up an error handler with LCMS2 that will throw a [LibraryError](#) exception.*
- Exiv2::ExifKey [exif\\_key\\_read](#) (std::string key\_string)

- Exiv2::Value::AutoPtr [exif\\_value\\_read](#) (Exiv2::ExifKey key, std::string value\_string)  
*Read an EXIF value from a string, with optional substitution for enum-style values.*
- Exiv2::IptcKey [iptc\\_key\\_read](#) (std::string key\_string)
- Exiv2::XmpKey [xmp\\_key\\_read](#) (std::string key\_string)
- template<typename Num\_type , typename R\_type >  
Exiv2::Value::AutoPtr [parse\\_Rational](#) (std::string s)  
*Parse a string into a rational fraction.*

## Variables

- bool [benchmark\\_mode](#) = false
- [jxr\\_format\\_subst](#) [JXR\\_format\\_table](#)
- unsigned char [header](#) [12]
- std::map< std::string, WebPPreset > [WebP\\_presets](#)
- [subst\\_table](#) [EXIF\\_key\\_subst](#)  
*Map from Image::Exiftool tag names to Exiv2's tag names.*
- std::map< std::string, [subst\\_table](#) > [EXIF\\_value\\_subst](#)
- [subst\\_table](#) [IPTC\\_key\\_subst](#)  
*Map from Image::Exiftool tag names to Exiv2's tag names.*
- [subst\\_table](#) [XMP\\_key\\_subst](#)  
*Map from Image::Exiftool tag names to Exiv2's tag names.*

### 5.2.1 Typedef Documentation

#### 5.2.1.1 hash

```
typedef std::map<std::string, std::string> PhotoFinish::hash
```

A simple hash.

Definition at line 36 of file Destination\_items.hh.

#### 5.2.1.2 jxr\_format\_subst

```
typedef std::vector<std::pair<unsigned int, const PKPixelFormatGUID*> > PhotoFinish::jxr\_↵  
format\_subst
```

Definition at line 34 of file JXR.hh.

### 5.2.1.3 multihash

```
typedef std::map<std::string, stringlist > PhotoFinish::multihash
```

A hash of string lists.

Definition at line 42 of file Destination\_items.hh.

### 5.2.1.4 rulerlist

```
typedef std::list< rulerpair > PhotoFinish::rulerlist
```

A list of rulers.

Definition at line 36 of file CropSolution.hh.

### 5.2.1.5 rulerpair

```
typedef std::pair<double, double> PhotoFinish::rulerpair
```

Ruler paramaters - percentage of final image vs. pixel position in original.

Definition at line 33 of file CropSolution.hh.

### 5.2.1.6 stringlist

```
typedef std::vector<std::string> PhotoFinish::stringlist
```

A list of strings.

Definition at line 39 of file Destination\_items.hh.

### 5.2.1.7 subst\_table

```
typedef std::vector<std::pair<std::string, std::string> > PhotoFinish::subst_table
```

Definition at line 37 of file Tags.hh.

## 5.2.2 Function Documentation



### 5.2.2.1 add\_ruler\_pins()

```
void PhotoFinish::add_ruler_pins (
    rulerlist & rulers,
    unsigned int max )
```

Add rulers to the either side of an image if there aren't enough.

Definition at line 53 of file CropSolution.cc.

### 5.2.2.2 add\_rulers()

```
void PhotoFinish::add_rulers (
    multihash & vars,
    std::string key,
    rulerlist & rulers )
```

Parse named variables into a list of rulers.

#### Parameters

<i>vars</i>	The tag variables
<i>key</i>	The name of the variables to parse
<i>rulers</i>	The list of rulers to add to

Definition at line 36 of file CropSolution.cc.

### 5.2.2.3 closest\_Rational()

```
template<typename Num_type , typename R_type >
Exiv2::ValueType<R_type>& PhotoFinish::closest_Rational (
    double value )
```

Find a close rational fraction given a floating-point value.

Definition at line 101 of file Tags.hh.

### 5.2.2.4 copy\_le\_to()

```
void PhotoFinish::copy_le_to (
    unsigned char * dest,
    unsigned int value,
    unsigned char length ) [inline]
```

Definition at line 95 of file WebP\_ostream.hh.

#### 5.2.2.5 error\_callback()

```
void PhotoFinish::error_callback (
    const char * msg,
    void * client_data )
```

Error callback for OpenJPEG - throw a [LibraryError](#) exception.

Definition at line 25 of file JP2\_callbacks.cc.

#### 5.2.2.6 exif\_key\_read()

```
Exiv2::ExifKey PhotoFinish::exif_key_read (
    std::string key_string )
```

Definition at line 81 of file Tags\_EXIF\_subst.cc.

#### 5.2.2.7 exif\_value\_read()

```
Exiv2::Value::AutoPtr PhotoFinish::exif_value_read (
    Exiv2::ExifKey key,
    std::string value_string )
```

Read an EXIF value from a string, with optional substitution for enum-style values.

Definition at line 280 of file Tags\_EXIF\_subst.cc.

#### 5.2.2.8 exists()

```
bool PhotoFinish::exists (
    const ImageFilepath & fp ) [inline]
```

Definition at line 92 of file ImageFile.hh.

#### 5.2.2.9 info\_callback()

```
void PhotoFinish::info_callback (
    const char * msg,
    void * client_data )
```

Info callback for OpenJPEG - print the indented message to STDERR.

Definition at line 34 of file JP2\_callbacks.cc.

#### 5.2.2.10 iptc\_key\_read()

```
Exiv2::IptcKey PhotoFinish::iptc_key_read (
    std::string key_string )
```

Definition at line 37 of file Tags\_IPTC\_subst.cc.

#### 5.2.2.11 jpeg\_error\_exit()

```
void PhotoFinish::jpeg_error_exit (
    j_common_ptr cinfo )
```

Definition at line 36 of file JPEGwriter.cc.

#### 5.2.2.12 jpeg\_istream\_fill\_input\_buffer()

```
boolean PhotoFinish::jpeg_istream_fill_input_buffer (
    j_decompress_ptr dinfo )
```

Fill the buffer.

Definition at line 45 of file JPEG\_istream.cc.

#### 5.2.2.13 jpeg\_istream\_init\_source()

```
void PhotoFinish::jpeg_istream_init_source (
    j_decompress_ptr dinfo )
```

Initialise the istream source manager.

Definition at line 34 of file JPEG\_istream.cc.

#### 5.2.2.14 jpeg\_istream\_resync\_to\_restart()

```
boolean PhotoFinish::jpeg_istream_resync_to_restart (
    j_decompress_ptr dinfo,
    int desired )
```

Resync to start?!?

Definition at line 74 of file JPEG\_istream.cc.

#### 5.2.2.15 jpeg\_istream\_skip\_input\_data()

```
void PhotoFinish::jpeg_istream_skip_input_data (
    j_decompress_ptr dinfo,
    long num_bytes )
```

Skip some data.

Definition at line 57 of file JPEG\_iostream.cc.

#### 5.2.2.16 jpeg\_istream\_src()

```
void PhotoFinish::jpeg_istream_src (
    j_decompress_ptr dinfo,
    std::istream * is )
```

Set up a "source manager" on the given JPEG decompression structure to read from an istream.

Definition at line 84 of file JPEG\_iostream.cc.

#### 5.2.2.17 jpeg\_istream\_src\_free()

```
void PhotoFinish::jpeg_istream_src_free (
    j_decompress_ptr dinfo )
```

Free the data structures of the istream source manager.

Definition at line 99 of file JPEG\_iostream.cc.

#### 5.2.2.18 jpeg\_istream\_term\_source()

```
void PhotoFinish::jpeg_istream_term_source (
    j_decompress_ptr dinfo )
```

Terminate the istream source manager.

Definition at line 79 of file JPEG\_iostream.cc.

#### 5.2.2.19 jpeg\_ostream\_dest()

```
void PhotoFinish::jpeg_ostream_dest (
    j_compress_ptr cinfo,
    std::ostream * os )
```

Setup a "destination manager" on the given JPEG compression structure to write to an ostream.

Definition at line 144 of file JPEG\_ostream.cc.

#### 5.2.2.20 jpeg\_ostream\_dest\_free()

```
void PhotoFinish::jpeg_ostream_dest_free (
    j_compress_ptr cinfo )
```

Free the data structures of the ostream destination manager.

Definition at line 157 of file JPEG\_ostream.cc.

#### 5.2.2.21 jpeg\_read\_profile()

```
CMS::Profile::ptr PhotoFinish::jpeg_read_profile (
    jpeg_decompress_struct * dinfo,
    Destination::ptr dest )
```

Read an ICC profile from APP2 markers in a JPEG file.

Definition at line 31 of file JPEG\_profiles.cc.

#### 5.2.2.22 jpeg\_write\_profile()

```
void PhotoFinish::jpeg_write_profile (
    jpeg_compress_struct * cinfo,
    unsigned char * data,
    unsigned int size )
```

Write an ICC profile into APP2 markers in a JPEG file.

Definition at line 78 of file JPEG\_profiles.cc.

#### 5.2.2.23 jpegfile\_scan\_greyscale()

```
void PhotoFinish::jpegfile_scan_greyscale (
    jpeg_compress_struct * cinfo )
```

Create a scan "script" for a greyscale image.

Create a scan script for encoding a greyscale progressive JPEG.

Definition at line 114 of file JPEG\_scans.cc.

#### 5.2.2.24 jpegfile\_scan\_RGB()

```
void PhotoFinish::jpegfile_scan_RGB (
    jpeg_compress_struct * cinfo )
```

Create a scan "script" for an RGB image.

Create a scan script for encoding a colour progressive JPEG.

Definition at line 26 of file JPEG\_scans.cc.

#### 5.2.2.25 jxr\_cms\_format()

```
CMS::Format PhotoFinish::jxr_cms_format (
    const PKPixelFormatGUID & g )
```

Definition at line 51 of file JXR\_formats.cc.

#### 5.2.2.26 jxr\_pixel\_format()

```
const PKPixelFormatGUID & PhotoFinish::jxr_pixel_format (
    unsigned int n )
```

Definition at line 43 of file JXR\_formats.cc.

#### 5.2.2.27 last\_write\_time()

```
std::time_t PhotoFinish::last_write_time (
    const ImageFilepath & fp ) [inline]
```

Definition at line 93 of file ImageFile.hh.

#### 5.2.2.28 lcms2\_error\_adaptor()

```
void PhotoFinish::lcms2_error_adaptor (
    void )
```

Set up an error handler with LCMS2 that will throw a [LibraryError](#) exception.

Definition at line 29 of file LCMS2ErrorHandler.cc.

#### 5.2.2.29 lcms2\_errorhandler()

```
void PhotoFinish::lcms2_errorhandler (
    cmsContext ContextID,
    cmsUInt32Number ErrorCode,
    const char * Text )
```

Throw a [LibraryError](#) exception when LCMS2 returns an error.

Definition at line 25 of file LCMS2ErrorHandler.cc.

#### 5.2.2.30 limitval()

```
template<typename T >
T PhotoFinish::limitval (
    SAMPLE v )
```

A template function that limits a floating-point value while converting to another type.

#### 5.2.2.31 limitval< double >()

```
template<>
double PhotoFinish::limitval< double > (
    SAMPLE v ) [inline]
```

Definition at line 250 of file Image.hh.

#### 5.2.2.32 limitval< float >()

```
template<>
float PhotoFinish::limitval< float > (
    SAMPLE v ) [inline]
```

Definition at line 245 of file Image.hh.

#### 5.2.2.33 `limitval< unsigned char >()`

```
template<>
unsigned char PhotoFinish::limitval< unsigned char > (
    SAMPLE v ) [inline]
```

Definition at line 209 of file Image.hh.

#### 5.2.2.34 `limitval< unsigned int >()`

```
template<>
unsigned int PhotoFinish::limitval< unsigned int > (
    SAMPLE v ) [inline]
```

Definition at line 227 of file Image.hh.

#### 5.2.2.35 `limitval< unsigned long long >()`

```
template<>
unsigned long long PhotoFinish::limitval< unsigned long long > (
    SAMPLE v ) [inline]
```

Definition at line 236 of file Image.hh.

#### 5.2.2.36 `limitval< unsigned short int >()`

```
template<>
unsigned short int PhotoFinish::limitval< unsigned short int > (
    SAMPLE v ) [inline]
```

Definition at line 218 of file Image.hh.

#### 5.2.2.37 `operator<<()`

```
std::ostream & PhotoFinish::operator<< (
    std::ostream & out,
    Timer t )
```

Definition at line 43 of file Benchmark.cc.



### 5.2.2.38 parse\_Rational()

```
template<typename Num_type , typename R_type >
Exiv2::Value::AutoPtr PhotoFinish::parse_Rational (
    std::string s )
```

Parse a string into a rational fraction.

Definition at line 267 of file Tags\_EXIF\_subst.cc.

### 5.2.2.39 png\_end\_cb()

```
void PhotoFinish::png_end_cb (
    png_structp png,
    png_infop info )
```

Called by libPNG when the image data has finished.

Definition at line 141 of file PNGreader\_cb.cc.

### 5.2.2.40 png\_flush\_ostream\_cb()

```
void PhotoFinish::png_flush_ostream_cb (
    png_structp png )
```

libPNG callback for flushing an ostream

Definition at line 65 of file PNGwriter.cc.

### 5.2.2.41 png\_info\_cb()

```
void PhotoFinish::png_info_cb (
    png_structp png,
    png_infop info )
```

Called by libPNG when the iHDR chunk has been read with the main "header" information.

Definition at line 121 of file PNGreader\_cb.cc.

#### 5.2.2.42 png\_row\_cb()

```
void PhotoFinish::png_row_cb (
    png_structp png,
    png_bytep row_data,
    png_uint_32 row_num,
    int pass )
```

Called by libPNG when a row of image data has been read.

Definition at line 132 of file PNGreader\_cb.cc.

#### 5.2.2.43 png\_write\_ostream\_cb()

```
void PhotoFinish::png_write_ostream_cb (
    png_structp png,
    png_bytep buffer,
    png_size_t length )
```

libPNG callback for writing to an ostream

Definition at line 59 of file PNGwriter.cc.

#### 5.2.2.44 profile\_name()

```
std::string PhotoFinish::profile_name (
    CMS::Profile::ptr profile )
```

Definition at line 143 of file Image.cc.

#### 5.2.2.45 read\_le32()

```
unsigned int PhotoFinish::read_le32 (
    const unsigned char * data ) [inline]
```

Definition at line 102 of file WebP\_ostream.hh.

#### 5.2.2.46 read\_planar()

```
template<typename T >
void PhotoFinish::read_planar (
    unsigned int width,
    unsigned char channels,
    opj_image_t * image,
    T * row,
    unsigned int y ) [inline]
```

Read a row of image data from OpenJPEG's planar integer components into an LCMS2-compatible single array.

Definition at line 36 of file JP2.hh.

#### 5.2.2.47 scaleval()

```
template<typename T >
T PhotoFinish::scaleval (
    void )
```

A template function that returns the 'scale' value of a type.

#### 5.2.2.48 scaleval< double >()

```
template<>
double PhotoFinish::scaleval< double > (
    void ) [inline]
```

Definition at line 201 of file Image.hh.

#### 5.2.2.49 scaleval< float >()

```
template<>
float PhotoFinish::scaleval< float > (
    void ) [inline]
```

Definition at line 198 of file Image.hh.

#### 5.2.2.50 scaleval< unsigned char >()

```
template<>
unsigned char PhotoFinish::scaleval< unsigned char > (
    void ) [inline]
```

Definition at line 186 of file Image.hh.

#### 5.2.2.51 `scaleval< unsigned int >()`

```
template<>
unsigned int PhotoFinish::scaleval< unsigned int > (
    void ) [inline]
```

Definition at line 192 of file Image.hh.

#### 5.2.2.52 `scaleval< unsigned long long >()`

```
template<>
unsigned long long PhotoFinish::scaleval< unsigned long long > (
    void ) [inline]
```

Definition at line 195 of file Image.hh.

#### 5.2.2.53 `scaleval< unsigned short int >()`

```
template<>
unsigned short int PhotoFinish::scaleval< unsigned short int > (
    void ) [inline]
```

Definition at line 189 of file Image.hh.

#### 5.2.2.54 `transfer_alpha()`

```
void PhotoFinish::transfer_alpha (
    unsigned int width,
    CMS::Format src_format,
    const unsigned char * src_row,
    CMS::Format dest_format,
    const unsigned char * dest_row )
```

Definition at line 118 of file Image.cc.

#### 5.2.2.55 `transfer_alpha_typed()`

```
template<typename A >
void PhotoFinish::transfer_alpha_typed (
    unsigned int width,
    unsigned char src_channels,
    const A * src_row,
    CMS::Format dest_format,
    const unsigned char * dest_row )
```

Definition at line 92 of file Image.cc.

#### 5.2.2.56 transfer\_alpha\_typed2()

```
template<typename A , typename B >
void PhotoFinish::transfer_alpha_typed2 (
    unsigned int width,
    unsigned char src_channels,
    const A * src_row,
    unsigned char dest_channels,
    const B * dest_row )
```

Definition at line 83 of file Image.cc.

#### 5.2.2.57 warning\_callback()

```
void PhotoFinish::warning_callback (
    const char * msg,
    void * client_data )
```

Warning callback for OpenJPEG - print the message to STDERR.

Definition at line 29 of file JP2\_callbacks.cc.

#### 5.2.2.58 webp\_stream\_writer\_func()

```
int PhotoFinish::webp_stream_writer_func (
    const uint8_t * data,
    size_t data_size,
    const WebPPicture * picture )
```

Wrapper around the [webp\\_stream\\_writer](#) class.

Definition at line 187 of file WebP\_ostream.cc.

#### 5.2.2.59 write\_be()

```
void PhotoFinish::write_be (
    void * ptr,
    size_t size,
    std::ostream & stream )
```

Definition at line 46 of file SOLwriter.cc.

#### 5.2.2.60 write\_packed()

```
template<typename T >
void PhotoFinish::write_packed (
    unsigned int width,
    unsigned char channels,
    T * row,
    opj_image_t * image,
    unsigned int y )
```

Read a row of packed pixel data into OpenJPEG's planar components.

Definition at line 60 of file JP2.hh.

#### 5.2.2.61 write\_planar()

```
template<typename T >
void PhotoFinish::write_planar (
    unsigned int width,
    unsigned char channels,
    T * row,
    opj_image_t * image,
    unsigned int y )
```

Read a row of planar pixel data into OpenJPEG's planar components.

Definition at line 48 of file JP2.hh.

#### 5.2.2.62 xmp\_key\_read()

```
Exiv2::XmpKey PhotoFinish::xmp_key_read (
    std::string key_string )
```

Definition at line 43 of file Tags\_XMP\_subst.cc.

### 5.2.3 Variable Documentation

#### 5.2.3.1 benchmark\_mode

```
bool PhotoFinish::benchmark_mode = false
```

Definition at line 23 of file Benchmark.cc.

## 5.2.3.2 EXIF\_key\_subst

```
subst_table PhotoFinish::EXIF_key_subst
```

Map from Image::Exiftool tag names to Exiv2's tag names.

Definition at line 27 of file Tags\_EXIF\_subst.cc.

## 5.2.3.3 EXIF\_value\_subst

```
std::map<std::string, subst_table> PhotoFinish::EXIF_value_subst
```

Definition at line 93 of file Tags\_EXIF\_subst.cc.

## 5.2.3.4 header

```
unsigned char PhotoFinish::header[12]
```

**Initial value:**

```
= { 0x53, 0x4f, 0x4c, 0x3a, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00 }
```

Definition at line 43 of file SOLwriter.cc.

## 5.2.3.5 IPTC\_key\_subst

```
subst_table PhotoFinish::IPTC_key_subst
```

**Initial value:**

```
= {
    StrPair("IPTC:By-line",          "Iptc.Application2.Byline"),
    StrPair("IPTC:City",            "Iptc.Application2.City"),
    StrPair("IPTC:Country-PrimaryLocationCode", "Iptc.Application2.CountryCode"),
    StrPair("IPTC:Country-PrimaryLocationName", "Iptc.Application2.CountryName"),
    StrPair("IPTC:CopyrightNotice",  "Iptc.Application2.Copyright"),
    StrPair("IPTC:Province-State",    "Iptc.Application2.ProvinceState"),
    StrPair("IPTC:Sub-location",      "Iptc.Application2.SubLocation"),
}
```

Map from Image::Exiftool tag names to Exiv2's tag names.

Definition at line 27 of file Tags\_IPTC\_subst.cc.

### 5.2.3.6 JXR\_format\_table

`jxr_format_subst` PhotoFinish::JXR\_format\_table

#### Initial value:

```
= {
    FmtPair(TYPE_GRAY_8, &GUID_PKPixelFormat8bppGray),
    FmtPair(TYPE_GRAY_16, &GUID_PKPixelFormat16bppGray),

    FmtPair(TYPE_BGR_8, &GUID_PKPixelFormat24bppBGR),
    FmtPair(TYPE_RGB_8, &GUID_PKPixelFormat24bppRGB),

    FmtPair(TYPE_BGRA_8, &GUID_PKPixelFormat32bppBGR),
    FmtPair(TYPE_BGRA_8, &GUID_PKPixelFormat32bppBGRA),

    FmtPair(TYPE_GRAY_FLT, &GUID_PKPixelFormat32bppGrayFloat),
    FmtPair(TYPE_RGBA_8, &GUID_PKPixelFormat32bppRGB),
    FmtPair(TYPE_RGBA_8, &GUID_PKPixelFormat32bppRGBA),
}
```

Definition at line 25 of file JXR\_formats.cc.

### 5.2.3.7 WebP\_presets

`std::map<std::string, WebPPreset>` PhotoFinish::WebP\_presets

#### Initial value:

```
= { std::make_pair("Default", WEBP_PRESET_DEFAULT),
    std::make_pair("Picture", WEBP_PRESET_PICTURE),
    std::make_pair("Photo", WEBP_PRESET_PHOTO),
    std::make_pair("Drawing", WEBP_PRESET_DRAWING),
    std::make_pair("Icon", WEBP_PRESET_ICON),
    std::make_pair("Text", WEBP_PRESET_TEXT) }
```

Definition at line 28 of file WebPwriter.cc.

### 5.2.3.8 XMP\_key\_subst

`subst_table` PhotoFinish::XMP\_key\_subst

#### Initial value:

```
= {
    StrPair("XMP:Copyright", "Xmp.dc.Copyright"),
    StrPair("XMP:Creator", "Xmp.dc.Creator"),

    StrPair("XMP:CreatorContactInfoCiAdrCity", "Xmp.iptc.CiAdrCity"),
    StrPair("XMP:CreatorContactInfoCiAdrCtry", "Xmp.iptc.CiAdrCtry"),
    StrPair("XMP:CreatorContactInfoCiAdrExtadr", "Xmp.iptc.CiAdrExtadr"),
    StrPair("XMP:CreatorContactInfoCiAdrPcode", "Xmp.iptc.CiAdrPcode"),

    StrPair("XMP-cc:License", "Xmp.cc.License"),

    StrPair("XMP-microsoft:CameraSerialNumber", "Xmp.MicrosoftPhoto.CameraSerialNumber"),
    StrPair("XMP-microsoft:LensManufacturer", "Xmp.MicrosoftPhoto.LensManufacturer"),
    StrPair("XMP-microsoft:LensModel", "Xmp.MicrosoftPhoto.LensModel"),
}
```

Map from Image::Exiftool tag names to Exiv2's tag names.

Definition at line 27 of file Tags\_XMP\_subst.cc.



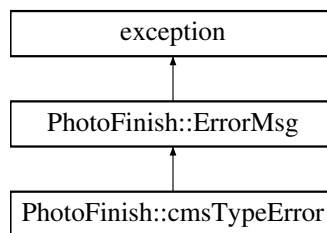
## Chapter 6

# Class Documentation

### 6.1 PhotoFinish::cmsTypeError Class Reference

```
#include <Exception.hh>
```

Inheritance diagram for PhotoFinish::cmsTypeError:



#### Public Member Functions

- [cmsTypeError](#) (const std::string &m, const unsigned int &t)  
*Constructor.*
- virtual const char \* [what](#) () const noexcept

#### Additional Inherited Members

##### 6.1.1 Detailed Description

Definition at line 302 of file Exception.hh.

##### 6.1.2 Constructor & Destructor Documentation

###### 6.1.2.1 cmsTypeError()

```
PhotoFinish::cmsTypeError::cmsTypeError (
    const std::string & m,
    const unsigned int & t ) [inline]
```

Constructor.

**Parameters**

<i>m</i>	Message string.
<i>t</i>	LCMS2 type.

Definition at line 312 of file Exception.hh.

**6.1.3 Member Function Documentation****6.1.3.1 what()**

```
virtual const char* PhotoFinish::cmsTypeError::what ( ) const [inline], [virtual], [noexcept]
```

Implements [PhotoFinish::ErrorMsg](#).

Definition at line 316 of file Exception.hh.

The documentation for this class was generated from the following file:

- [Exception.hh](#)

**6.2 PhotoFinish::CropSolver Class Reference**

Class for finding the best frame position for cropping.

```
#include <CropSolution.hh>
```

**Public Member Functions**

- [CropSolver](#) ([multihash](#) &vars)
- [Frame::ptr solve](#) ([Image::ptr](#) img, [D\\_target::ptr](#) target)

**6.2.1 Detailed Description**

Class for finding the best frame position for cropping.

Definition at line 39 of file CropSolution.hh.

**6.2.2 Constructor & Destructor Documentation**

### 6.2.2.1 CropSolver()

```
PhotoFinish::CropSolver::CropSolver (
    multihash & vars )
```

Definition at line 47 of file CropSolution.cc.

## 6.2.3 Member Function Documentation

### 6.2.3.1 solve()

```
Frame::ptr PhotoFinish::CropSolver::solve (
    Image::ptr img,
    D_target::ptr target )
```

Definition at line 68 of file CropSolution.cc.

The documentation for this class was generated from the following files:

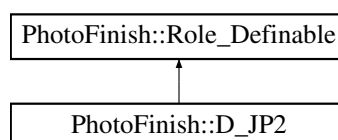
- [CropSolution.hh](#)
- [CropSolution.cc](#)

## 6.3 PhotoFinish::D\_JP2 Class Reference

JP2 parameters for destination.

```
#include <Destination_items.hh>
```

Inheritance diagram for PhotoFinish::D\_JP2:



## Public Member Functions

- [D\\_JP2](#) ()  
*Empty constructor.*
- void [add\\_variables](#) ([multihash](#) &vars)  
*Set values from a map of "variables".*
- [definable](#)< int > [numresolutions](#) (void) const
- void [set\\_numresolutions](#) (int n)
- [definable](#)< std::string > [prog\\_order](#) (void) const
- void [set\\_prog\\_order](#) (const std::string &po)
- int [num\\_rates](#) (void) const
- float [rate](#) (int n) const
- void [set\\_rate](#) (int n, float r)
- void [set\\_rates](#) (std::vector< float > r)
- int [num\\_qualities](#) (void) const
- float [quality](#) (int n) const
- void [set\\_quality](#) (int n, float r)
- void [set\\_qualities](#) (std::vector< float > r)
- [definable](#)< std::pair< int, int > > [tile\\_size](#) (void) const
- void [set\\_tile\\_size](#) (int h, int v)
- [definable](#)< bool > [reversible](#) (void) const
- void [set\\_reversible](#) (bool r=true)
- void [set\\_irreversible](#) (void)
- void [read\\_config](#) (const YAML::Node &node)  
*Read a [D\\_JP2](#) record from a YAML file.*

## Additional Inherited Members

### 6.3.1 Detailed Description

JP2 parameters for destination.

Definition at line 180 of file Destination\_items.hh.

### 6.3.2 Constructor & Destructor Documentation

#### 6.3.2.1 D\_JP2()

```
PhotoFinish::D_JP2::D_JP2 ( )
```

Empty constructor.

Definition at line 218 of file Destination\_items.cc.

### 6.3.3 Member Function Documentation

#### 6.3.3.1 add\_variables()

```
void PhotoFinish::D_JP2::add_variables (
    multihash & vars )
```

Set values from a map of "variables".

Definition at line 223 of file Destination\_items.cc.

#### 6.3.3.2 num\_qualities()

```
int PhotoFinish::D_JP2::num_qualities (
    void ) const [inline]
```

Definition at line 206 of file Destination\_items.hh.

#### 6.3.3.3 num\_rates()

```
int PhotoFinish::D_JP2::num_rates (
    void ) const [inline]
```

Definition at line 201 of file Destination\_items.hh.

#### 6.3.3.4 numresolutions()

```
definable<int> PhotoFinish::D_JP2::numresolutions (
    void ) const [inline]
```

Definition at line 195 of file Destination\_items.hh.

#### 6.3.3.5 prog\_order()

```
definable<std::string> PhotoFinish::D_JP2::prog_order (
    void ) const [inline]
```

Definition at line 198 of file Destination\_items.hh.

#### 6.3.3.6 quality()

```
float PhotoFinish::D_JP2::quality (
    int n ) const [inline]
```

Definition at line 207 of file Destination\_items.hh.

#### 6.3.3.7 rate()

```
float PhotoFinish::D_JP2::rate (
    int n ) const [inline]
```

Definition at line 202 of file Destination\_items.hh.

#### 6.3.3.8 read\_config()

```
void PhotoFinish::D_JP2::read_config (
    const YAML::Node & node )
```

Read a [D\\_JP2](#) record from a YAML file.

Definition at line 319 of file Destination\_items.cc.

#### 6.3.3.9 reversible()

```
definable<bool> PhotoFinish::D_JP2::reversible (
    void ) const [inline]
```

Definition at line 214 of file Destination\_items.hh.

#### 6.3.3.10 set\_irreversible()

```
void PhotoFinish::D_JP2::set_irreversible (
    void ) [inline]
```

Definition at line 216 of file Destination\_items.hh.

**6.3.3.11 set\_numresolutions()**

```
void PhotoFinish::D_JP2::set_numresolutions (
    int n ) [inline]
```

Definition at line 196 of file Destination\_items.hh.

**6.3.3.12 set\_prog\_order()**

```
void PhotoFinish::D_JP2::set_prog_order (
    const std::string & po ) [inline]
```

Definition at line 199 of file Destination\_items.hh.

**6.3.3.13 set\_qualities()**

```
void PhotoFinish::D_JP2::set_qualities (
    std::vector< float > r ) [inline]
```

Definition at line 209 of file Destination\_items.hh.

**6.3.3.14 set\_quality()**

```
void PhotoFinish::D_JP2::set_quality (
    int n,
    float r ) [inline]
```

Definition at line 208 of file Destination\_items.hh.

**6.3.3.15 set\_rate()**

```
void PhotoFinish::D_JP2::set_rate (
    int n,
    float r ) [inline]
```

Definition at line 203 of file Destination\_items.hh.

**6.3.3.16 set\_rates()**

```
void PhotoFinish::D_JP2::set_rates (
    std::vector< float > r ) [inline]
```

Definition at line 204 of file Destination\_items.hh.

**6.3.3.17 set\_reversible()**

```
void PhotoFinish::D_JP2::set_reversible (
    bool r = true ) [inline]
```

Definition at line 215 of file Destination\_items.hh.

**6.3.3.18 set\_tile\_size()**

```
void PhotoFinish::D_JP2::set_tile_size (
    int h,
    int v ) [inline]
```

Definition at line 212 of file Destination\_items.hh.

**6.3.3.19 tile\_size()**

```
definable< std::pair<int, int> > PhotoFinish::D_JP2::tile_size (
    void ) const [inline]
```

Definition at line 211 of file Destination\_items.hh.

The documentation for this class was generated from the following files:

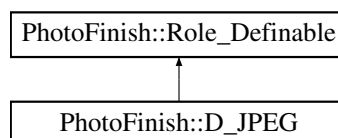
- [Destination\\_items.hh](#)
- [Destination\\_items.cc](#)

**6.4 PhotoFinish::D\_JPEG Class Reference**

JPEG parameters for destination.

```
#include <Destination_items.hh>
```

Inheritance diagram for PhotoFinish::D\_JPEG:





## Public Member Functions

- [D\\_JPEG](#) ()  
*Empty constructor.*
- [D\\_JPEG](#) (int q, char h, char v, bool p)  
*Constructor.*
- void [add\\_variables](#) (multihash &vars)  
*Set values from a map of "variables".*
- [definable](#)< int > [quality](#) (void) const
- void [set\\_quality](#) (int q)
- [definable](#)< std::pair< int, int > > [sample](#) (void) const
- void [set\\_sample](#) (int h, int v)
- [definable](#)< bool > [progressive](#) (void) const
- void [set\\_progressive](#) (bool p=true)
- void [read\\_config](#) (const YAML::Node &node)  
*Read a [D\\_JPEG](#) record from a YAML file.*

## Additional Inherited Members

### 6.4.1 Detailed Description

JPEG parameters for destination.

Definition at line 105 of file Destination\_items.hh.

### 6.4.2 Constructor & Destructor Documentation

#### 6.4.2.1 D\_JPEG() [1/2]

```
PhotoFinish::D_JPEG::D_JPEG ( )
```

Empty constructor.

Definition at line 91 of file Destination\_items.cc.

#### 6.4.2.2 D\_JPEG() [2/2]

```
PhotoFinish::D_JPEG::D_JPEG (
    int q,
    char h,
    char v,
    bool p )
```

Constructor.

## Parameters

<i>q</i>	Quality
<i>h,v</i>	Chroma sampling
<i>p</i>	Progressive

Definition at line 94 of file Destination\_items.cc.

### 6.4.3 Member Function Documentation

#### 6.4.3.1 add\_variables()

```
void PhotoFinish::D_JPEG::add_variables (
    multihash & vars )
```

Set values from a map of "variables".

Definition at line 100 of file Destination\_items.cc.

#### 6.4.3.2 progressive()

```
definable<bool> PhotoFinish::D_JPEG::progressive (
    void ) const [inline]
```

Definition at line 132 of file Destination\_items.hh.

#### 6.4.3.3 quality()

```
definable<int> PhotoFinish::D_JPEG::quality (
    void ) const [inline]
```

Definition at line 126 of file Destination\_items.hh.

#### 6.4.3.4 read\_config()

```
void PhotoFinish::D_JPEG::read_config (
    const YAML::Node & node )
```

Read a [D\\_JPEG](#) record from a YAML file.

Definition at line 131 of file Destination\_items.cc.

#### 6.4.3.5 sample()

```
definable< std::pair<int, int> > PhotoFinish::D_JPEG::sample (
    void ) const [inline]
```

Definition at line 129 of file Destination\_items.hh.

#### 6.4.3.6 set\_progressive()

```
void PhotoFinish::D_JPEG::set_progressive (
    bool p = true ) [inline]
```

Definition at line 133 of file Destination\_items.hh.

#### 6.4.3.7 set\_quality()

```
void PhotoFinish::D_JPEG::set_quality (
    int q ) [inline]
```

Definition at line 127 of file Destination\_items.hh.

#### 6.4.3.8 set\_sample()

```
void PhotoFinish::D_JPEG::set_sample (
    int h,
    int v ) [inline]
```

Definition at line 130 of file Destination\_items.hh.

The documentation for this class was generated from the following files:

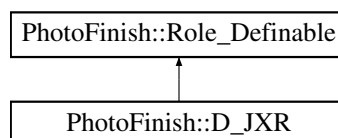
- [Destination\\_items.hh](#)
- [Destination\\_items.cc](#)

## 6.5 PhotoFinish::D\_JXR Class Reference

JPEG XR parameters for destination.

```
#include <Destination_items.hh>
```

Inheritance diagram for PhotoFinish::D\_JXR:



## Public Member Functions

- [D\\_JXR](#) ()  
*Empty constructor.*
- void [add\\_variables](#) ([multihash](#) &vars)  
*Set values from a map of "variables".*
- int [quality](#) (void) const
- void [set\\_quality](#) (int iq)
- int [alphaq](#) (void) const
- void [set\\_alphaq](#) (int aq)
- [definable](#)< std::string > [overlap](#) (void) const
- void [set\\_overlap](#) (const std::string &o)
- std::string [subsampling](#) (void) const
- void [set\\_subsampling](#) (const std::string &s)
- [definable](#)< int > [tilesize](#) (void) const
- void [set\\_tilesize](#) (int ts)
- [definable](#)< bool > [progressive](#) (void) const
- void [set\\_progressive](#) (bool p=true)
- void [set\\_sequential](#) (bool s=true)
- void [read\\_config](#) (const YAML::Node &node)

## Additional Inherited Members

### 6.5.1 Detailed Description

JPEG XR parameters for destination.

Definition at line 254 of file Destination\_items.hh.

### 6.5.2 Constructor & Destructor Documentation

#### 6.5.2.1 D\_JXR()

```
PhotoFinish::D_JXR::D_JXR ( )
```

Empty constructor.

Definition at line 416 of file Destination\_items.cc.

### 6.5.3 Member Function Documentation

#### 6.5.3.1 add\_variables()

```
void PhotoFinish::D_JXR::add_variables (
    multihash & vars )
```

Set values from a map of "variables".

Definition at line 422 of file Destination\_items.cc.

#### 6.5.3.2 alphaq()

```
int PhotoFinish::D_JXR::alphaq (
    void ) const [inline]
```

Definition at line 272 of file Destination\_items.hh.

#### 6.5.3.3 overlap()

```
definable<std::string> PhotoFinish::D_JXR::overlap (
    void ) const [inline]
```

Definition at line 275 of file Destination\_items.hh.

#### 6.5.3.4 progressive()

```
definable<bool> PhotoFinish::D_JXR::progressive (
    void ) const [inline]
```

Definition at line 284 of file Destination\_items.hh.

#### 6.5.3.5 quality()

```
int PhotoFinish::D_JXR::quality (
    void ) const [inline]
```

Definition at line 269 of file Destination\_items.hh.

#### 6.5.3.6 read\_config()

```
void PhotoFinish::D_JXR::read_config (
    const YAML::Node & node )
```

Definition at line 425 of file Destination\_items.cc.

#### 6.5.3.7 set\_alphag()

```
void PhotoFinish::D_JXR::set_alphag (
    int ag ) [inline]
```

Definition at line 273 of file Destination\_items.hh.

#### 6.5.3.8 set\_overlap()

```
void PhotoFinish::D_JXR::set_overlap (
    const std::string & o ) [inline]
```

Definition at line 276 of file Destination\_items.hh.

#### 6.5.3.9 set\_progressive()

```
void PhotoFinish::D_JXR::set_progressive (
    bool p = true ) [inline]
```

Definition at line 285 of file Destination\_items.hh.

#### 6.5.3.10 set\_quality()

```
void PhotoFinish::D_JXR::set_quality (
    int iq ) [inline]
```

Definition at line 270 of file Destination\_items.hh.

#### 6.5.3.11 set\_sequential()

```
void PhotoFinish::D_JXR::set_sequential (
    bool s = true ) [inline]
```

Definition at line 286 of file Destination\_items.hh.

**6.5.3.12 set\_subsampling()**

```
void PhotoFinish::D_JXR::set_subsampling (
    const std::string & s ) [inline]
```

Definition at line 279 of file Destination\_items.hh.

**6.5.3.13 set\_tilesize()**

```
void PhotoFinish::D_JXR::set_tilesize (
    int ts ) [inline]
```

Definition at line 282 of file Destination\_items.hh.

**6.5.3.14 subsampling()**

```
std::string PhotoFinish::D_JXR::subsampling (
    void ) const [inline]
```

Definition at line 278 of file Destination\_items.hh.

**6.5.3.15 tileSize()**

```
definable<int> PhotoFinish::D_JXR::tilesize (
    void ) const [inline]
```

Definition at line 281 of file Destination\_items.hh.

The documentation for this class was generated from the following files:

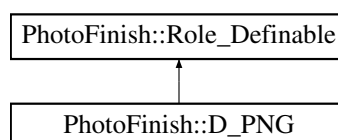
- [Destination\\_items.hh](#)
- [Destination\\_items.cc](#)

**6.6 PhotoFinish::D\_PNG Class Reference**

PNG parameters for destination.

```
#include <Destination_items.hh>
```

Inheritance diagram for PhotoFinish::D\_PNG:



## Public Member Functions

- [D\\_PNG](#) ()
- void [read\\_config](#) (const YAML::Node &node)  
*Read a [D\\_PNG](#) record from a YAML file.*

## Additional Inherited Members

### 6.6.1 Detailed Description

PNG parameters for destination.

Definition at line 139 of file Destination\_items.hh.

### 6.6.2 Constructor & Destructor Documentation

#### 6.6.2.1 D\_PNG()

```
PhotoFinish::D_PNG::D_PNG ( )
```

Definition at line 158 of file Destination\_items.cc.

### 6.6.3 Member Function Documentation

#### 6.6.3.1 read\_config()

```
void PhotoFinish::D_PNG::read_config (
    const YAML::Node & node )
```

Read a [D\\_PNG](#) record from a YAML file.

Definition at line 162 of file Destination\_items.cc.

The documentation for this class was generated from the following files:

- [Destination\\_items.hh](#)
- [Destination\\_items.cc](#)

## 6.7 PhotoFinish::D\_profile Class Reference

ICC profile parameters for destination.

```
#include <Destination_items.hh>
```



## Public Types

- typedef std::shared\_ptr< [D\\_profile](#) > [ptr](#)  
*Shared pointer for a [D\\_profile](#).*

## Public Member Functions

- [D\\_profile](#) ()  
*Empty constructor.*
- [D\\_profile](#) (const std::string &[name](#), fs::path [filepath](#))  
*Constructor.*
- [D\\_profile](#) (const std::string &[name](#), unsigned char \*[data](#), unsigned int [data\\_size](#))  
*Constructor.*
- [D\\_profile](#) (const [D\\_profile](#) &[other](#))  
*Copy constructor.*
- ~[D\\_profile](#) ()  
*Destructor.*
- [D\\_profile](#) & [operator=](#) (const [D\\_profile](#) &[b](#))  
*Assignment operator.*
- [definable](#)< std::string > [name](#) (void) const  
*Name of the profile.*
- [definable](#)< fs::path > [filepath](#) (void) const  
*File path for reading the profile.*
- bool [has\\_data](#) (void) const  
*Do we have the profile data instead of a file path?*
- [CMS::Profile::ptr](#) [profile](#) (void) const  
*The profile data for LCMS2.*
- unsigned char \* [data](#) (void) const  
*The profile data.*
- unsigned int [data\\_size](#) (void) const  
*The size of the profile data.*
- void [read\\_config](#) (const YAML::Node &[node](#))  
*Read a [D\\_profile](#) record from a YAML file.*

### 6.7.1 Detailed Description

ICC profile parameters for destination.

Definition at line 292 of file Destination\_items.hh.

### 6.7.2 Member Typedef Documentation

#### 6.7.2.1 ptr

```
typedef std::shared_ptr<D\_profile> PhotoFinish::D\_profile::ptr
```

Shared pointer for a [D\\_profile](#).

Definition at line 339 of file Destination\_items.hh.

### 6.7.3 Constructor & Destructor Documentation

#### 6.7.3.1 D\_profile() [1/4]

```
PhotoFinish::D_profile::D_profile ( )
```

Empty constructor.

Definition at line 450 of file Destination\_items.cc.

#### 6.7.3.2 D\_profile() [2/4]

```
PhotoFinish::D_profile::D_profile (
    const std::string & name,
    fs::path filepath )
```

Constructor.

Definition at line 454 of file Destination\_items.cc.

#### 6.7.3.3 D\_profile() [3/4]

```
PhotoFinish::D_profile::D_profile (
    const std::string & name,
    unsigned char * data,
    unsigned int data_size )
```

Constructor.

Definition at line 460 of file Destination\_items.cc.

#### 6.7.3.4 D\_profile() [4/4]

```
PhotoFinish::D_profile::D_profile (
    const D_profile & other )
```

Copy constructor.

Definition at line 465 of file Destination\_items.cc.

#### 6.7.3.5 ~D\_profile()

```
PhotoFinish::D_profile::~~D_profile ( )
```

Destructor.

Definition at line 475 of file Destination\_items.cc.

### 6.7.4 Member Function Documentation

#### 6.7.4.1 data()

```
unsigned char* PhotoFinish::D_profile::data (
    void ) const [inline]
```

The profile data.

Definition at line 331 of file Destination\_items.hh.

#### 6.7.4.2 data\_size()

```
unsigned int PhotoFinish::D_profile::data_size (
    void ) const [inline]
```

The size of the profile data.

Definition at line 334 of file Destination\_items.hh.

#### 6.7.4.3 filepath()

```
definable<fs::path> PhotoFinish::D_profile::filepath (
    void ) const [inline]
```

File path for reading the profile.

Definition at line 322 of file Destination\_items.hh.

#### 6.7.4.4 has\_data()

```
bool PhotoFinish::D_profile::has_data (
    void ) const [inline]
```

Do we have the profile data instead of a file path?

Definition at line 325 of file Destination\_items.hh.

#### 6.7.4.5 name()

```
definable<std::string> PhotoFinish::D_profile::name (
    void ) const [inline]
```

Name of the profile.

Definition at line 319 of file Destination\_items.hh.

#### 6.7.4.6 operator=()

```
D_profile & PhotoFinish::D_profile::operator= (
    const D_profile & b )
```

Assignment operator.

Definition at line 483 of file Destination\_items.cc.

#### 6.7.4.7 profile()

```
CMS::Profile::ptr PhotoFinish::D_profile::profile (
    void ) const
```

The profile data for LCMS2.

Definition at line 495 of file Destination\_items.cc.

#### 6.7.4.8 read\_config()

```
void PhotoFinish::D_profile::read_config (
    const YAML::Node & node )
```

Read a [D\\_profile](#) record from a YAML file.

Definition at line 505 of file Destination\_items.cc.

The documentation for this class was generated from the following files:

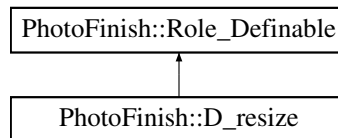
- [Destination\\_items.hh](#)
- [Destination\\_items.cc](#)

## 6.8 PhotoFinish::D\_resize Class Reference

Resize parameters for destination.

```
#include <Destination_items.hh>
```

Inheritance diagram for PhotoFinish::D\_resize:



### Public Member Functions

- [D\\_resize](#) ()  
*Empty constructor.*
- [definable](#)< std::string > [filter](#) (void) const
- [definable](#)< double > [support](#) (void) const
- void [read\\_config](#) (const YAML::Node &node)  
*Read a [D\\_resize](#) record from a YAML file.*

### Static Public Member Functions

- static [D\\_resize lanczos](#) (double r)  
*Named constructor.*

### Additional Inherited Members

#### 6.8.1 Detailed Description

Resize parameters for destination.

Definition at line 60 of file Destination\_items.hh.

#### 6.8.2 Constructor & Destructor Documentation

##### 6.8.2.1 D\_resize()

```
PhotoFinish::D_resize::D_resize ( )
```

Empty constructor.

Definition at line 48 of file Destination\_items.cc.

## 6.8.3 Member Function Documentation

### 6.8.3.1 filter()

```
definable<std::string> PhotoFinish::D_resize::filter (
    void ) const [inline]
```

Definition at line 77 of file Destination\_items.hh.

### 6.8.3.2 lanczos()

```
static D_resize PhotoFinish::D_resize::lanczos (
    double r ) [inline], [static]
```

Named constructor.

Constructs a [D\\_resize](#) object with filter="lanczos" and the supplied radius

Parameters

<i>r</i>	Radius of <a href="#">Lanczos</a> filter
----------	--

Definition at line 75 of file Destination\_items.hh.

### 6.8.3.3 read\_config()

```
void PhotoFinish::D_resize::read_config (
    const YAML::Node & node )
```

Read a [D\\_resize](#) record from a YAML file.

Definition at line 56 of file Destination\_items.cc.

### 6.8.3.4 support()

```
definable<double> PhotoFinish::D_resize::support (
    void ) const [inline]
```

Definition at line 78 of file Destination\_items.hh.

The documentation for this class was generated from the following files:

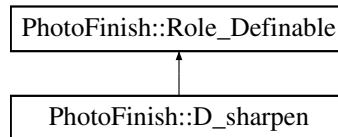
- [Destination\\_items.hh](#)
- [Destination\\_items.cc](#)

## 6.9 PhotoFinish::D\_sharpen Class Reference

Sharpen parameters for destination.

```
#include <Destination_items.hh>
```

Inheritance diagram for PhotoFinish::D\_sharpen:



### Public Member Functions

- [D\\_sharpen](#) ()  
*Empty constructor.*
- [definable](#)< double > [radius](#) (void) const
- [definable](#)< double > [sigma](#) (void) const
- void [read\\_config](#) (const YAML::Node &node)  
*Read a [D\\_sharpen](#) record from a YAML file.*

### Additional Inherited Members

#### 6.9.1 Detailed Description

Sharpen parameters for destination.

Definition at line 45 of file Destination\_items.hh.

#### 6.9.2 Constructor & Destructor Documentation

##### 6.9.2.1 D\_sharpen()

```
PhotoFinish::D_sharpen::D_sharpen ( )
```

Empty constructor.

Definition at line 32 of file Destination\_items.cc.

#### 6.9.3 Member Function Documentation

### 6.9.3.1 radius()

```
definable<double> PhotoFinish::D_sharpen::radius (
    void ) const [inline]
```

Definition at line 53 of file Destination\_items.hh.

### 6.9.3.2 read\_config()

```
void PhotoFinish::D_sharpen::read_config (
    const YAML::Node & node )
```

Read a [D\\_sharpen](#) record from a YAML file.

Definition at line 36 of file Destination\_items.cc.

### 6.9.3.3 sigma()

```
definable<double> PhotoFinish::D_sharpen::sigma (
    void ) const [inline]
```

Definition at line 54 of file Destination\_items.hh.

The documentation for this class was generated from the following files:

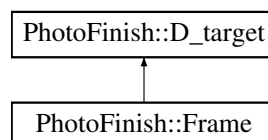
- [Destination\\_items.hh](#)
- [Destination\\_items.cc](#)

## 6.10 PhotoFinish::D\_target Class Reference

Target parameters for destination.

```
#include <Destination_items.hh>
```

Inheritance diagram for PhotoFinish::D\_target:



### Public Types

- typedef std::shared\_ptr< [D\\_target](#) > [ptr](#)



## Public Member Functions

- [D\\_target](#) (const std::string &n, double w, double h)  
*A target-specific size (in inches) to override the one in the destination.*
- [D\\_target](#) (const std::string &n)
- std::string [name](#) (void) const
- [definable](#)< double > [width](#) (void) const
- [definable](#)< double > [height](#) (void) const
- [definable](#)< double > [size](#) (void) const
- void [read\\_config](#) (const YAML::Node &node)  
*Read a [D\\_target](#) record from a YAML file.*

## Protected Attributes

- std::string [\\_name](#)
- [definable](#)< double > [\\_width](#)
- [definable](#)< double > [\\_height](#)
- [definable](#)< double > [\\_size](#)

### 6.10.1 Detailed Description

Target parameters for destination.

Definition at line 84 of file Destination\_items.hh.

### 6.10.2 Member Typedef Documentation

#### 6.10.2.1 ptr

```
typedef std::shared_ptr<D\_target> PhotoFinish::D\_target::ptr
```

Definition at line 101 of file Destination\_items.hh.

### 6.10.3 Constructor & Destructor Documentation

#### 6.10.3.1 D\_target() [1/2]

```
PhotoFinish::D_target::D_target (
    const std::string & n,
    double w,
    double h )
```

A target-specific size (in inches) to override the one in the destination.

Definition at line 68 of file Destination\_items.cc.

#### 6.10.3.2 `D_target()` [2/2]

```
PhotoFinish::D_target::D_target (
    const std::string & n )
```

Definition at line 73 of file `Destination_items.cc`.

### 6.10.4 Member Function Documentation

#### 6.10.4.1 `height()`

```
definable<double> PhotoFinish::D_target::height (
    void ) const [inline]
```

Definition at line 96 of file `Destination_items.hh`.

#### 6.10.4.2 `name()`

```
std::string PhotoFinish::D_target::name (
    void ) const [inline]
```

Definition at line 94 of file `Destination_items.hh`.

#### 6.10.4.3 `read_config()`

```
void PhotoFinish::D_target::read_config (
    const YAML::Node & node )
```

Read a `D_target` record from a YAML file.

Definition at line 78 of file `Destination_items.cc`.

#### 6.10.4.4 `size()`

```
definable<double> PhotoFinish::D_target::size (
    void ) const [inline]
```

Definition at line 97 of file `Destination_items.hh`.

#### 6.10.4.5 width()

```
definable<double> PhotoFinish::D_target::width (  
    void ) const [inline]
```

Definition at line 95 of file Destination\_items.hh.

### 6.10.5 Member Data Documentation

#### 6.10.5.1 \_height

```
definable<double> PhotoFinish::D_target::_height [protected]
```

Definition at line 87 of file Destination\_items.hh.

#### 6.10.5.2 \_name

```
std::string PhotoFinish::D_target::_name [protected]
```

Definition at line 86 of file Destination\_items.hh.

#### 6.10.5.3 \_size

```
definable<double> PhotoFinish::D_target::_size [protected]
```

Definition at line 88 of file Destination\_items.hh.

#### 6.10.5.4 \_width

```
definable<double> PhotoFinish::D_target::_width [protected]
```

Definition at line 87 of file Destination\_items.hh.

The documentation for this class was generated from the following files:

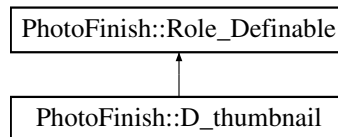
- [Destination\\_items.hh](#)
- [Destination\\_items.cc](#)

## 6.11 PhotoFinish::D\_thumbnail Class Reference

Thumbnail parameters for destination.

```
#include <Destination_items.hh>
```

Inheritance diagram for PhotoFinish::D\_thumbnail:



### Public Member Functions

- [D\\_thumbnail](#) ()
- [definable](#)< bool > [generate](#) (void) const
- [definable](#)< double > [maxwidth](#) (void) const
- [definable](#)< double > [maxheight](#) (void) const
- void [read\\_config](#) (const YAML::Node &node)

*Read a [D\\_thumbnail](#) record from a YAML file.*

### Additional Inherited Members

#### 6.11.1 Detailed Description

Thumbnail parameters for destination.

Definition at line 343 of file Destination\_items.hh.

#### 6.11.2 Constructor & Destructor Documentation

##### 6.11.2.1 D\_thumbnail()

```
PhotoFinish::D_thumbnail::D_thumbnail ( )
```

Definition at line 514 of file Destination\_items.cc.

#### 6.11.3 Member Function Documentation

## 6.11.3.1 generate()

```
definable<bool> PhotoFinish::D_thumbnail::generate (
    void ) const [inline]
```

Definition at line 351 of file Destination\_items.hh.

## 6.11.3.2 maxheight()

```
definable<double> PhotoFinish::D_thumbnail::maxheight (
    void ) const [inline]
```

Definition at line 353 of file Destination\_items.hh.

## 6.11.3.3 maxwidth()

```
definable<double> PhotoFinish::D_thumbnail::maxwidth (
    void ) const [inline]
```

Definition at line 352 of file Destination\_items.hh.

## 6.11.3.4 read\_config()

```
void PhotoFinish::D_thumbnail::read_config (
    const YAML::Node & node )
```

Read a [D\\_thumbnail](#) record from a YAML file.

Definition at line 518 of file Destination\_items.cc.

The documentation for this class was generated from the following files:

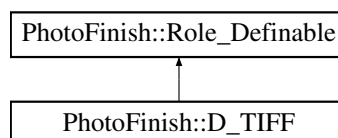
- [Destination\\_items.hh](#)
- [Destination\\_items.cc](#)

## 6.12 PhotoFinish::D\_TIFF Class Reference

TIFF parameters for destination.

```
#include <Destination_items.hh>
```

Inheritance diagram for PhotoFinish::D\_TIFF:



## Public Member Functions

- [D\\_TIFF](#) ()  
*Empty constructor.*
- [D\\_TIFF](#) (const std::string &c)  
*Constructor.*
- void [add\\_variables](#) (multihash &vars)  
*Set values from a map of "variables".*
- [definable](#)< std::string > [artist](#) (void) const
- void [set\\_artist](#) (const std::string &a)
- [definable](#)< std::string > [copyright](#) (void) const
- void [set\\_copyright](#) (const std::string &c)
- [definable](#)< std::string > [compression](#) (void) const
- void [set\\_compression](#) (const std::string &c)
- void [read\\_config](#) (const YAML::Node &node)  
*Read a [D\\_TIFF](#) record from a YAML file.*

## Additional Inherited Members

### 6.12.1 Detailed Description

TIFF parameters for destination.

Definition at line 149 of file Destination\_items.hh.

### 6.12.2 Constructor & Destructor Documentation

#### 6.12.2.1 [D\\_TIFF\(\)](#) [1/2]

```
PhotoFinish::D_TIFF::D_TIFF ( )
```

Empty constructor.

Definition at line 167 of file Destination\_items.cc.

#### 6.12.2.2 [D\\_TIFF\(\)](#) [2/2]

```
PhotoFinish::D_TIFF::D_TIFF (
    const std::string & c )
```

Constructor.

#### Parameters

c	Compression string
---	--------------------

Definition at line 170 of file Destination\_items.cc.

### 6.12.3 Member Function Documentation

#### 6.12.3.1 add\_variables()

```
void PhotoFinish::D_TIFF::add_variables (
    multihash & vars )
```

Set values from a map of "variables".

Definition at line 174 of file Destination\_items.cc.

#### 6.12.3.2 artist()

```
definable<std::string> PhotoFinish::D_TIFF::artist (
    void ) const [inline]
```

Definition at line 167 of file Destination\_items.hh.

#### 6.12.3.3 compression()

```
definable<std::string> PhotoFinish::D_TIFF::compression (
    void ) const [inline]
```

Definition at line 173 of file Destination\_items.hh.

#### 6.12.3.4 copyright()

```
definable<std::string> PhotoFinish::D_TIFF::copyright (
    void ) const [inline]
```

Definition at line 170 of file Destination\_items.hh.

#### 6.12.3.5 read\_config()

```
void PhotoFinish::D_TIFF::read_config (
    const YAML::Node & node )
```

Read a [D\\_TIFF](#) record from a YAML file.

Definition at line 202 of file Destination\_items.cc.

#### 6.12.3.6 set\_artist()

```
void PhotoFinish::D_TIFF::set_artist (
    const std::string & a ) [inline]
```

Definition at line 168 of file Destination\_items.hh.

#### 6.12.3.7 set\_compression()

```
void PhotoFinish::D_TIFF::set_compression (
    const std::string & c ) [inline]
```

Definition at line 174 of file Destination\_items.hh.

#### 6.12.3.8 set\_copyright()

```
void PhotoFinish::D_TIFF::set_copyright (
    const std::string & c ) [inline]
```

Definition at line 171 of file Destination\_items.hh.

The documentation for this class was generated from the following files:

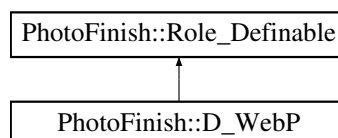
- [Destination\\_items.hh](#)
- [Destination\\_items.cc](#)

## 6.13 PhotoFinish::D\_WebP Class Reference

WebP parameters for destination.

```
#include <Destination_items.hh>
```

Inheritance diagram for PhotoFinish::D\_WebP:





## Public Member Functions

- [D\\_WebP](#) ()  
*Empty constructor.*
- void [add\\_variables](#) ([multihash](#) &vars)  
*Set values from a map of "variables".*
- [definable](#)< std::string > [preset](#) (void) const
- void [set\\_preset](#) (const std::string &p)
- [definable](#)< bool > [lossless](#) (void) const
- [definable](#)< bool > [lossy](#) (void) const
- void [set\\_lossless](#) (bool l=true)
- void [set\\_lossy](#) (bool l=true)
- float [quality](#) (void) const
- void [set\\_quality](#) (float q)
- [definable](#)< unsigned char > [method](#) (void) const
- void [set\\_method](#) (unsigned char m)
- void [read\\_config](#) (const YAML::Node &node)

## Additional Inherited Members

### 6.13.1 Detailed Description

WebP parameters for destination.

Definition at line 222 of file Destination\_items.hh.

### 6.13.2 Constructor & Destructor Documentation

#### 6.13.2.1 D\_WebP()

```
PhotoFinish::D_WebP::D_WebP ( )
```

Empty constructor.

Definition at line 390 of file Destination\_items.cc.

### 6.13.3 Member Function Documentation

#### 6.13.3.1 add\_variables()

```
void PhotoFinish::D_WebP::add_variables (
    multihash & vars )
```

Set values from a map of "variables".

Definition at line 394 of file Destination\_items.cc.

#### 6.13.3.2 lossless()

```
definable<bool> PhotoFinish::D_WebP::lossless (  
    void ) const [inline]
```

Definition at line 239 of file Destination\_items.hh.

#### 6.13.3.3 lossy()

```
definable<bool> PhotoFinish::D_WebP::lossy (  
    void ) const [inline]
```

Definition at line 240 of file Destination\_items.hh.

#### 6.13.3.4 method()

```
definable<unsigned char> PhotoFinish::D_WebP::method (  
    void ) const [inline]
```

Definition at line 247 of file Destination\_items.hh.

#### 6.13.3.5 preset()

```
definable<std::string> PhotoFinish::D_WebP::preset (  
    void ) const [inline]
```

Definition at line 236 of file Destination\_items.hh.

#### 6.13.3.6 quality()

```
float PhotoFinish::D_WebP::quality (  
    void ) const [inline]
```

Definition at line 244 of file Destination\_items.hh.

#### 6.13.3.7 read\_config()

```
void PhotoFinish::D_WebP::read_config (  
    const YAML::Node & node )
```

Definition at line 397 of file Destination\_items.cc.

#### 6.13.3.8 set\_lossless()

```
void PhotoFinish::D_WebP::set_lossless (
    bool l = true ) [inline]
```

Definition at line 241 of file Destination\_items.hh.

#### 6.13.3.9 set\_lossy()

```
void PhotoFinish::D_WebP::set_lossy (
    bool l = true ) [inline]
```

Definition at line 242 of file Destination\_items.hh.

#### 6.13.3.10 set\_method()

```
void PhotoFinish::D_WebP::set_method (
    unsigned char m ) [inline]
```

Definition at line 248 of file Destination\_items.hh.

#### 6.13.3.11 set\_preset()

```
void PhotoFinish::D_WebP::set_preset (
    const std::string & p ) [inline]
```

Definition at line 237 of file Destination\_items.hh.

#### 6.13.3.12 set\_quality()

```
void PhotoFinish::D_WebP::set_quality (
    float q ) [inline]
```

Definition at line 245 of file Destination\_items.hh.

The documentation for this class was generated from the following files:

- [Destination\\_items.hh](#)
- [Destination\\_items.cc](#)

## 6.14 PhotoFinish::definable< T > Class Template Reference

Template class for storing things that can be defined or undefined.

```
#include <Definable.hh>
```

### Public Member Functions

- [definable](#) ()  
*Empty constructor.*
- [definable](#) (const T &i)  
*Construct from an item.*
- const bool [defined](#) (void) const  
*Is this object defined?*
- void [set\\_defined](#) (bool v=true)  
*Set this object as 'defined' (or not)*
- void [undefine](#) (void)  
*Undefine the object.*
- T [get](#) (void)  
*Get the item.*
- const T & [get](#) (void) const  
*Get the item, const version.*
- [operator T](#) (void) const  
*Cast to the contained type.*
- T \* [operator->](#) ()  
*Arrow operator.*
- const T \* [operator->](#) () const  
*Arrow operator, const version.*
- [definable](#)< T > & [operator=](#) (const T &i)  
*Assignment operator.*

### Friends

- std::ostream & [operator<<](#) (std::ostream &out, [definable](#)< T > &data)  
*Allow the contained data to be output to an ostream.*

#### 6.14.1 Detailed Description

```
template<typename T>
class PhotoFinish::definable< T >
```

Template class for storing things that can be defined or undefined.

Definition at line 29 of file Definable.hh.

#### 6.14.2 Constructor & Destructor Documentation

#### 6.14.2.1 definable() [1/2]

```
template<typename T>
PhotoFinish::definable< T >::definable ( ) [inline]
```

Empty constructor.

This sets the object to undefined and the item is initialised with its empty constructor

Definition at line 39 of file Definable.hh.

#### 6.14.2.2 definable() [2/2]

```
template<typename T>
PhotoFinish::definable< T >::definable (
    const T & i ) [inline]
```

Construct from an item.

This obviously also sets the object to 'defined'

Definition at line 48 of file Definable.hh.

### 6.14.3 Member Function Documentation

#### 6.14.3.1 defined()

```
template<typename T>
const bool PhotoFinish::definable< T >::defined (
    void ) const [inline]
```

Is this object defined?

Definition at line 54 of file Definable.hh.

#### 6.14.3.2 get() [1/2]

```
template<typename T>
T PhotoFinish::definable< T >::get (
    void ) [inline]
```

Get the item.

Definition at line 65 of file Definable.hh.

#### 6.14.3.3 `get()` [2/2]

```
template<typename T>
const T& PhotoFinish::definable< T >::get (
    void ) const [inline]
```

Get the item, const version.

Definition at line 67 of file Definable.hh.

#### 6.14.3.4 `operator T()`

```
template<typename T>
PhotoFinish::definable< T >::operator T (
    void ) const [inline]
```

Cast to the contained type.

Definition at line 70 of file Definable.hh.

#### 6.14.3.5 `operator->()` [1/2]

```
template<typename T>
T* PhotoFinish::definable< T >::operator-> ( ) [inline]
```

Arrow operator.

Definition at line 73 of file Definable.hh.

#### 6.14.3.6 `operator->()` [2/2]

```
template<typename T>
const T* PhotoFinish::definable< T >::operator-> ( ) const [inline]
```

Arrow operator, const version.

Definition at line 75 of file Definable.hh.

#### 6.14.3.7 operator=()

```
template<typename T>
definable<T>& PhotoFinish::definable< T >::operator= (
    const T & i ) [inline]
```

Assignment operator.

Definition at line 78 of file Definable.hh.

#### 6.14.3.8 set\_defined()

```
template<typename T>
void PhotoFinish::definable< T >::set_defined (
    bool v = true ) [inline]
```

Set this object as 'defined' (or not)

Definition at line 59 of file Definable.hh.

#### 6.14.3.9 undefine()

```
template<typename T>
void PhotoFinish::definable< T >::undefine (
    void ) [inline]
```

Undefine the object.

Definition at line 62 of file Definable.hh.

### 6.14.4 Friends And Related Function Documentation

#### 6.14.4.1 operator<<

```
template<typename T>
std::ostream& operator<< (
    std::ostream & out,
    definable< T > & data ) [friend]
```

Allow the contained data to be output to an ostream.

Outputs "[undefined]" if the value is not defined.

Definition at line 88 of file Definable.hh.

The documentation for this class was generated from the following file:

- [Definable.hh](#)

## 6.15 PhotoFinish::Destination Class Reference

Represents a destination, read from destinations.yml.

```
#include <Destination.hh>
```

### Public Types

- typedef std::shared\_ptr< [Destination](#) > [ptr](#)  
*Shared pointer for a [Destination](#).*

### Public Member Functions

- [Destination](#) ()  
*Empty constructor.*
- [Destination](#) (const [Destination](#) &other)  
*Copy constructor.*
- [~Destination](#) ()  
*Destructor.*
- [Destination](#) & [operator=](#) (const [Destination](#) &b)  
*Assignment operator.*
- [ptr dupe](#) (void)  
*Duplicate.*
- [ptr add\\_variables](#) ([multihash](#) &vars)  
*Duplicate the current object and incorporate variables.*
- [Frame::ptr best\\_frame](#) ([Image::ptr](#) img)  
*Find the best crop+rescaling frame for an image.*
- [definable](#)< std::string > [name](#) (void) const
- const [definable](#)< fs::path > & [dir](#) (void) const
- [definable](#)< double > [size](#) (void) const
- const [D\\_sharpen](#) & [sharpen](#) (void) const
- const [D\\_resize](#) & [resize](#) (void) const
- int [num\\_targets](#) (void) const
- bool [has\\_targets](#) (void) const
- const std::map< std::string, [D\\_target::ptr](#) > & [targets](#) (void) const
- [definable](#)< std::string > [format](#) (void) const
- [definable](#)< int > [depth](#) (void) const
- void [set\\_depth](#) (int d)
- [definable](#)< bool > [noresize](#) (void) const
- [D\\_JPEG](#) & [jpeg](#) (void)
- void [set\\_jpeg](#) (const [D\\_JPEG](#) &j)
- [D\\_PNG](#) & [png](#) (void)
- void [set\\_png](#) (const [D\\_PNG](#) &p)
- [D\\_TIFF](#) & [tiff](#) (void)
- void [set\\_tiff](#) (const [D\\_TIFF](#) &t)
- [D\\_JP2](#) & [jp2](#) (void)
- void [set\\_jp2](#) (const [D\\_JP2](#) &j)
- [D\\_WebP](#) & [webp](#) (void)
- void [set\\_webp](#) (const [D\\_WebP](#) &w)
- [D\\_JXR](#) & [jxr](#) (void)
- void [set\\_jxr](#) (const [D\\_JXR](#) &j)



- `definable< CMS::Intent > intent` (void) const
- `CMS::Format modify_format` (CMS::Format format)  
*Modify an LCMS2 pixel format using some of the parameters in the destination.*
- `CMS::Profile::ptr get_profile` (CMS::ColourModel default\_colourmodel, std::string for\_desc)  
*Return an LCMS2 profile object from the profile data.*
- `const D_profile::ptr profile` (void) const
- `void set_profile` (std::string name, fs::path filepath)
- `void set_profile` (std::string name, unsigned char \*data, unsigned int data\_size)
- `void clear_profile` (void)
- `definable< bool > forcergb` (void) const
- `definable< bool > forcegrey` (void) const
- `const D_thumbnail & thumbnail` (void) const
- `void read_config` (const YAML::Node &node)  
*Read a destination record from a YAML document.*

### 6.15.1 Detailed Description

Represents a destination, read from destinations.yml.

Definition at line 37 of file Destination.hh.

### 6.15.2 Member Typedef Documentation

#### 6.15.2.1 ptr

```
typedef std::shared_ptr<Destination> PhotoFinish::Destination::ptr
```

Shared pointer for a [Destination](#).

Definition at line 87 of file Destination.hh.

### 6.15.3 Constructor & Destructor Documentation

#### 6.15.3.1 Destination() [1/2]

```
PhotoFinish::Destination::Destination ( )
```

Empty constructor.

Definition at line 34 of file Destination.cc.

#### 6.15.3.2 Destination() [2/2]

```
PhotoFinish::Destination::Destination (
    const Destination & other )
```

Copy constructor.

Definition at line 37 of file Destination.cc.

#### 6.15.3.3 ~Destination()

```
PhotoFinish::Destination::~~Destination ( )
```

Destructor.

Definition at line 58 of file Destination.cc.

### 6.15.4 Member Function Documentation

#### 6.15.4.1 add\_variables()

```
Destination::ptr PhotoFinish::Destination::add_variables (
    multihash & vars )
```

Duplicate the current object and incorporate variables.

Definition at line 91 of file Destination.cc.

#### 6.15.4.2 best\_frame()

```
Frame::ptr PhotoFinish::Destination::best_frame (
    Image::ptr img )
```

Find the best crop+rescaling frame for an image.

Definition at line 100 of file Destination.cc.

#### 6.15.4.3 clear\_profile()

```
void PhotoFinish::Destination::clear_profile (  
    void ) [inline]
```

Definition at line 148 of file Destination.hh.

#### 6.15.4.4 depth()

```
definable<int> PhotoFinish::Destination::depth (  
    void ) const [inline]
```

Definition at line 114 of file Destination.hh.

#### 6.15.4.5 dir()

```
const definable<fs::path>& PhotoFinish::Destination::dir (  
    void ) const [inline]
```

Definition at line 100 of file Destination.hh.

#### 6.15.4.6 dupe()

```
ptr PhotoFinish::Destination::dupe (  
    void ) [inline]
```

Duplicate.

Definition at line 90 of file Destination.hh.

#### 6.15.4.7 forcegrey()

```
definable<bool> PhotoFinish::Destination::forcegrey (  
    void ) const [inline]
```

Definition at line 151 of file Destination.hh.

#### 6.15.4.8 forcergb()

```
definable<bool> PhotoFinish::Destination::forcergb (  
    void ) const [inline]
```

Definition at line 150 of file Destination.hh.

#### 6.15.4.9 format()

```
definable<std::string> PhotoFinish::Destination::format (  
    void ) const [inline]
```

Definition at line 112 of file Destination.hh.

#### 6.15.4.10 get\_profile()

```
CMS::Profile::ptr PhotoFinish::Destination::get_profile (  
    CMS::ColourModel default_colourmodel,  
    std::string for_desc )
```

Return an LCMS2 profile object from the profile data.

Definition at line 190 of file Destination.cc.

#### 6.15.4.11 has\_targets()

```
bool PhotoFinish::Destination::has_targets (  
    void ) const [inline]
```

Definition at line 109 of file Destination.hh.

#### 6.15.4.12 intent()

```
definable<CMS::Intent> PhotoFinish::Destination::intent (  
    void ) const [inline]
```

Definition at line 137 of file Destination.hh.

#### 6.15.4.13 jp2()

```
D_JP2& PhotoFinish::Destination::jp2 (  
    void ) [inline]
```

Definition at line 128 of file Destination.hh.

#### 6.15.4.14 jpeg()

```
D_JPEG& PhotoFinish::Destination::jpeg (  
    void ) [inline]
```

Definition at line 119 of file Destination.hh.

#### 6.15.4.15 jxr()

```
D_JXR& PhotoFinish::Destination::jxr (  
    void ) [inline]
```

Definition at line 134 of file Destination.hh.

#### 6.15.4.16 modify\_format()

```
CMS::Format PhotoFinish::Destination::modify_format (  
    CMS::Format format )
```

Modify an LCMS2 pixel format using some of the parameters in the destination.

Definition at line 154 of file Destination.cc.

#### 6.15.4.17 name()

```
definable<std::string> PhotoFinish::Destination::name (  
    void ) const [inline]
```

Definition at line 98 of file Destination.hh.

#### 6.15.4.18 noresize()

```
definable<bool> PhotoFinish::Destination::noresize (
    void ) const [inline]
```

Definition at line 117 of file Destination.hh.

#### 6.15.4.19 num\_targets()

```
int PhotoFinish::Destination::num_targets (
    void ) const [inline]
```

Definition at line 108 of file Destination.hh.

#### 6.15.4.20 operator=()

```
Destination & PhotoFinish::Destination::operator= (
    const Destination & b )
```

Assignment operator.

Definition at line 61 of file Destination.cc.

#### 6.15.4.21 png()

```
D_PNG& PhotoFinish::Destination::png (
    void ) [inline]
```

Definition at line 122 of file Destination.hh.

#### 6.15.4.22 profile()

```
const D_profile::ptr PhotoFinish::Destination::profile (
    void ) const [inline]
```

Definition at line 145 of file Destination.hh.

#### 6.15.4.23 read\_config()

```
void PhotoFinish::Destination::read_config (
    const YAML::Node & node )
```

Read a destination record from a YAML document.

Read a [Destination](#) record from a YAML file.

Definition at line 207 of file Destination.cc.

#### 6.15.4.24 resize()

```
const D_resize& PhotoFinish::Destination::resize (
    void ) const [inline]
```

Definition at line 106 of file Destination.hh.

#### 6.15.4.25 set\_depth()

```
void PhotoFinish::Destination::set_depth (
    int d ) [inline]
```

Definition at line 115 of file Destination.hh.

#### 6.15.4.26 set\_jp2()

```
void PhotoFinish::Destination::set_jp2 (
    const D_JP2 & j ) [inline]
```

Definition at line 129 of file Destination.hh.

#### 6.15.4.27 set\_jpeg()

```
void PhotoFinish::Destination::set_jpeg (
    const D_JPEG & j ) [inline]
```

Definition at line 120 of file Destination.hh.

**6.15.4.28 set\_jxr()**

```
void PhotoFinish::Destination::set_jxr (
    const D_JXR & j ) [inline]
```

Definition at line 135 of file Destination.hh.

**6.15.4.29 set\_png()**

```
void PhotoFinish::Destination::set_png (
    const D_PNG & p ) [inline]
```

Definition at line 123 of file Destination.hh.

**6.15.4.30 set\_profile()** [1/2]

```
void PhotoFinish::Destination::set_profile (
    std::string name,
    fs::path filepath ) [inline]
```

Definition at line 146 of file Destination.hh.

**6.15.4.31 set\_profile()** [2/2]

```
void PhotoFinish::Destination::set_profile (
    std::string name,
    unsigned char * data,
    unsigned int data_size ) [inline]
```

Definition at line 147 of file Destination.hh.

**6.15.4.32 set\_tiff()**

```
void PhotoFinish::Destination::set_tiff (
    const D_TIFF & t ) [inline]
```

Definition at line 126 of file Destination.hh.



#### 6.15.4.33 set\_webp()

```
void PhotoFinish::Destination::set_webp (  
    const D_WebP & w ) [inline]
```

Definition at line 132 of file Destination.hh.

#### 6.15.4.34 sharpen()

```
const D_sharpen& PhotoFinish::Destination::sharpen (  
    void ) const [inline]
```

Definition at line 104 of file Destination.hh.

#### 6.15.4.35 size()

```
definable<double> PhotoFinish::Destination::size (  
    void ) const [inline]
```

Definition at line 102 of file Destination.hh.

#### 6.15.4.36 targets()

```
const std::map<std::string, D_target::ptr>& PhotoFinish::Destination::targets (  
    void ) const [inline]
```

Definition at line 110 of file Destination.hh.

#### 6.15.4.37 thumbnail()

```
const D_thumbnail& PhotoFinish::Destination::thumbnail (  
    void ) const [inline]
```

Definition at line 153 of file Destination.hh.

#### 6.15.4.38 tiff()

```
D_TIFF& PhotoFinish::Destination::tiff (  
    void ) [inline]
```

Definition at line 125 of file Destination.hh.

#### 6.15.4.39 webp()

```
D_WebP& PhotoFinish::Destination::webp (
    void ) [inline]
```

Definition at line 131 of file Destination.hh.

The documentation for this class was generated from the following files:

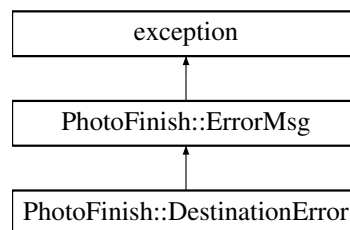
- [Destination.hh](#)
- [Destination.cc](#)

## 6.16 PhotoFinish::DestinationError Class Reference

[Destination](#) exception.

```
#include <Exception.hh>
```

Inheritance diagram for PhotoFinish::DestinationError:



### Public Member Functions

- [DestinationError](#) (const std::string &p, const std::string &v)  
*Constructor.*
- virtual const char \* [what](#) () const noexcept

### Additional Inherited Members

#### 6.16.1 Detailed Description

[Destination](#) exception.

Definition at line 263 of file Exception.hh.

#### 6.16.2 Constructor & Destructor Documentation

##### 6.16.2.1 DestinationError()

```
PhotoFinish::DestinationError::DestinationError (
    const std::string & p,
    const std::string & v ) [inline]
```

Constructor.

## Parameters

<i>p</i>	<a href="#">Destination</a> field "path"
<i>v</i>	Value that is wrong

Definition at line 273 of file Exception.hh.

### 6.16.3 Member Function Documentation

#### 6.16.3.1 what()

```
virtual const char* PhotoFinish::DestinationError::what ( ) const [inline], [virtual], [noexcept]
```

Implements [PhotoFinish::ErrorMsg](#).

Definition at line 277 of file Exception.hh.

The documentation for this class was generated from the following file:

- [Exception.hh](#)

## 6.17 PhotoFinish::Destinations Class Reference

A wrapper class for reading destinations from a YAML file and storing them in a map.

```
#include <Destination.hh>
```

### Public Types

- typedef std::map< std::string, [Destination::ptr](#) >::iterator iterator  
*Iterator for stepping through destinations.*
- typedef std::map< std::string, [Destination::ptr](#) >::const\_iterator const\_iterator  
*Constant iterator for stepping through destinations.*

### Public Member Functions

- [Destinations](#) (fs::path filepath)
- [Destinations](#) (const [Destinations](#) &other)
- [~Destinations](#) ()
- [Destinations](#) & [operator=](#) (const [Destinations](#) &b)
- void [Load](#) (fs::path filepath)
- std::map< std::string, [Destination::ptr](#) >::size\_type [count](#) (const std::string &key) const
- [iterator begin](#) (void)
- [const\\_iterator begin](#) (void) const
- [iterator end](#) (void)
- [const\\_iterator end](#) (void) const
- [Destination::ptr operator\[\]](#) (const std::string &key)

## Friends

- [iterator begin](#) ([Destinations](#) &d)
- [iterator end](#) ([Destinations](#) &d)

### 6.17.1 Detailed Description

A wrapper class for reading destinations from a YAML file and storing them in a map.

Definition at line 160 of file `Destination.hh`.

### 6.17.2 Member Typedef Documentation

#### 6.17.2.1 `const_iterator`

```
typedef std::map<std::string, Destination::ptr>::const_iterator PhotoFinish::Destinations↵  
::const_iterator
```

Constant iterator for stepping through destinations.

Definition at line 175 of file `Destination.hh`.

#### 6.17.2.2 `iterator`

```
typedef std::map<std::string, Destination::ptr>::iterator PhotoFinish::Destinations::iterator
```

Iterator for stepping through destinations.

Definition at line 172 of file `Destination.hh`.

### 6.17.3 Constructor & Destructor Documentation

#### 6.17.3.1 `Destinations()` [1/2]

```
PhotoFinish::Destinations::Destinations (  
    fs::path filepath )
```

Definition at line 291 of file `Destination.cc`.

### 6.17.3.2 Destinations() [2/2]

```
PhotoFinish::Destinations::Destinations (
    const Destinations & other )
```

Definition at line 295 of file Destination.cc.

### 6.17.3.3 ~Destinations()

```
PhotoFinish::Destinations::~Destinations ( )
```

Definition at line 300 of file Destination.cc.

## 6.17.4 Member Function Documentation

### 6.17.4.1 begin() [1/2]

```
iterator PhotoFinish::Destinations::begin (
    void ) [inline]
```

Definition at line 181 of file Destination.hh.

### 6.17.4.2 begin() [2/2]

```
const_iterator PhotoFinish::Destinations::begin (
    void ) const [inline]
```

Definition at line 182 of file Destination.hh.

### 6.17.4.3 count()

```
std::map<std::string, Destination::ptr>::size_type PhotoFinish::Destinations::count (
    const std::string & key ) const [inline]
```

Definition at line 179 of file Destination.hh.

#### 6.17.4.4 end() [1/2]

```
iterator PhotoFinish::Destinations::end (
    void ) [inline]
```

Definition at line 184 of file Destination.hh.

#### 6.17.4.5 end() [2/2]

```
const_iterator PhotoFinish::Destinations::end (
    void ) const [inline]
```

Definition at line 185 of file Destination.hh.

#### 6.17.4.6 Load()

```
void PhotoFinish::Destinations::Load (
    fs::path filepath )
```

Definition at line 312 of file Destination.cc.

#### 6.17.4.7 operator=()

```
Destinations & PhotoFinish::Destinations::operator= (
    const Destinations & b )
```

Definition at line 303 of file Destination.cc.

#### 6.17.4.8 operator[]()

```
Destination::ptr PhotoFinish::Destinations::operator[] (
    const std::string & key ) [inline]
```

Definition at line 193 of file Destination.hh.

### 6.17.5 Friends And Related Function Documentation

## 6.17.5.1 begin

```
iterator begin (
    Destinations & d ) [friend]
```

Definition at line 187 of file Destination.hh.

## 6.17.5.2 end

```
iterator end (
    Destinations & d ) [friend]
```

Definition at line 190 of file Destination.hh.

The documentation for this class was generated from the following files:

- [Destination.hh](#)
- [Destination.cc](#)

## 6.18 PhotoFinish::Ditherer Class Reference

Class for dithering images down to 8-bit components.

```
#include <Ditherer.hh>
```

### Public Member Functions

- [Ditherer](#) (unsigned int width, unsigned char channels, std::vector< unsigned char > maxvalues={})  
*Constructor.*
- [~Ditherer](#) ()  
*Destructor.*
- void [dither](#) (short unsigned int \*inrow, unsigned char \*outrow, bool lastrow=false)  
*Dither a row of image data.*

### Static Public Attributes

- static const cmsUInt32Number [cmsBaseType](#) = BYTES\_SH(2)  
*Base LCMS2 base type the ditherer expects the pixels to be in.*

### 6.18.1 Detailed Description

Class for dithering images down to 8-bit components.

Definition at line 27 of file Ditherer.hh.

## 6.18.2 Constructor & Destructor Documentation

### 6.18.2.1 Ditherer()

```
PhotoFinish::Ditherer::Ditherer (
    unsigned int width,
    unsigned char channels,
    std::vector< unsigned char > maxvalues = {} )
```

Constructor.

#### Parameters

<i>width</i>	Width of the image
<i>channels</i>	Number of channels of the image
<i>maxvalues</i>	The maximum values for each channel, defaults to 255 for each

Definition at line 28 of file Ditherer.cc.

### 6.18.2.2 ~Ditherer()

```
PhotoFinish::Ditherer::~~Ditherer ( )
```

Destructor.

Definition at line 49 of file Ditherer.cc.

## 6.18.3 Member Function Documentation

### 6.18.3.1 dither()

```
void PhotoFinish::Ditherer::dither (
    short unsigned int * inrow,
    unsigned char * outrow,
    bool lastrow = false )
```

Dither a row of image data.

Performs a Floyd-Steinberg error diffusion dither



## Parameters

<i>inrow</i>	Pointer to a row of 16-bit image data
<i>outrow</i>	Pointer to a row 8-bit image data that will be produced
<i>lastrow</i>	Whether this is the last row of the image. Less has to be done.

Definition at line 87 of file Ditherer.cc.

## 6.18.4 Member Data Documentation

### 6.18.4.1 cmsBaseType

```
const cmsUInt32Number PhotoFinish::Ditherer::cmsBaseType = BYTES_SH(2) [static]
```

Base LCMS2 base type the ditherer expects the pixels to be in.

Users of this class need to add the colour space and number of channels to this base type to be useable.

Definition at line 54 of file Ditherer.hh.

The documentation for this class was generated from the following files:

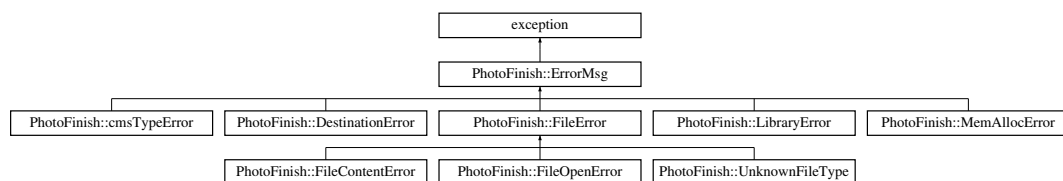
- [Ditherer.hh](#)
- [Ditherer.cc](#)

## 6.19 PhotoFinish::ErrorMsg Class Reference

Generic error message exception.

```
#include <Exception.hh>
```

Inheritance diagram for PhotoFinish::ErrorMsg:



## Public Member Functions

- [ErrorMsg](#) (const std::string &m)  
*Constructor.*
- virtual const char \* [what](#) () const noexcept=0

## Protected Attributes

- `const std::string _msg`

### 6.19.1 Detailed Description

Generic error message exception.

Definition at line 117 of file Exception.hh.

## 6.19.2 Constructor & Destructor Documentation

### 6.19.2.1 `ErrorMsg()`

```
PhotoFinish::ErrorMsg::ErrorMsg (
    const std::string & m ) [inline]
```

Constructor.

Parameters

<i>m</i>	Error message
----------	---------------

Definition at line 126 of file Exception.hh.

## 6.19.3 Member Function Documentation

### 6.19.3.1 `what()`

```
virtual const char* PhotoFinish::ErrorMsg::what ( ) const [pure virtual], [noexcept]
```

Implemented in [PhotoFinish::cmsTypeError](#), [PhotoFinish::LibraryError](#), [PhotoFinish::DestinationError](#), [PhotoFinish::FileContentError](#), [PhotoFinish::FileOpenError](#), [PhotoFinish::UnknownFileType](#), [PhotoFinish::FileError](#), and [PhotoFinish::MemAllocError](#).

## 6.19.4 Member Data Documentation

## 6.19.4.1 \_msg

```
const std::string PhotoFinish::ErrorMsg::_msg [protected]
```

Definition at line 119 of file Exception.hh.

The documentation for this class was generated from the following file:

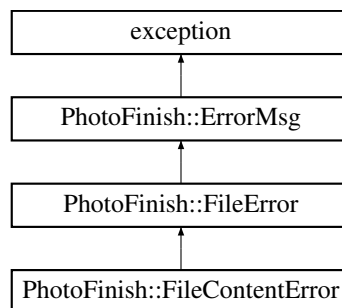
- [Exception.hh](#)

## 6.20 PhotoFinish::FileContentError Class Reference

File content exception.

```
#include <Exception.hh>
```

Inheritance diagram for PhotoFinish::FileContentError:



### Public Member Functions

- [FileContentError](#) (const std::string &fp, const std::string &m)  
*Constructor.*
- [FileContentError](#) (const std::string &fp)  
*Constructor.*
- virtual const char \* [what](#) () const noexcept

### Additional Inherited Members

#### 6.20.1 Detailed Description

File content exception.

Definition at line 234 of file Exception.hh.

#### 6.20.2 Constructor & Destructor Documentation

##### 6.20.2.1 FileContentError() [1/2]

```
PhotoFinish::FileContentError::FileContentError (
    const std::string & fp,
    const std::string & m ) [inline]
```

Constructor.

**Parameters**

<i>fp</i>	File path
<i>m</i>	Error message

Definition at line 241 of file Exception.hh.

**6.20.2.2 FileContentError()** [2/2]

```
PhotoFinish::FileContentError::FileContentError (
    const std::string & fp ) [inline]
```

Constructor.

**Parameters**

<i>fp</i>	File path
-----------	-----------

Definition at line 249 of file Exception.hh.

**6.20.3 Member Function Documentation****6.20.3.1 what()**

```
virtual const char* PhotoFinish::FileContentError::what ( ) const [inline], [virtual], [noexcept]
```

Implements [PhotoFinish::FileError](#).

Definition at line 253 of file Exception.hh.

The documentation for this class was generated from the following file:

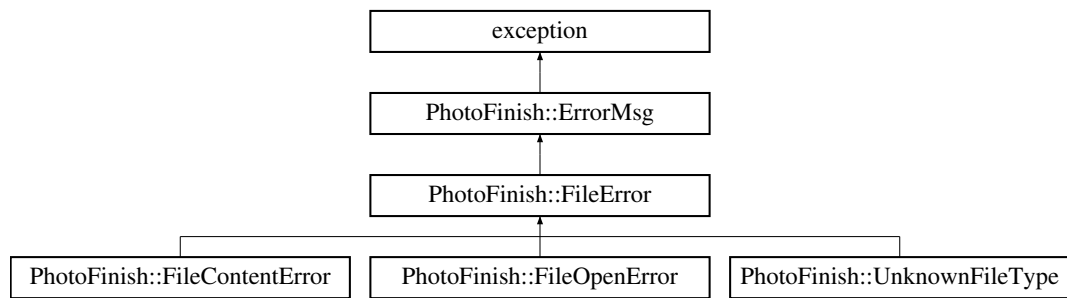
- [Exception.hh](#)

**6.21 PhotoFinish::FileError Class Reference**

File error abstract base exception.

```
#include <Exception.hh>
```

Inheritance diagram for PhotoFinish::FileError:



## Public Member Functions

- [FileError](#) (const std::string &fp, const std::string &m)  
*Constructor.*
- [FileError](#) (const std::string &fp)  
*Constructor.*
- virtual const char \* [what](#) () const noexcept=0

## Protected Attributes

- const std::string [\\_filepath](#)

### 6.21.1 Detailed Description

File error abstract base exception.

Definition at line 150 of file Exception.hh.

### 6.21.2 Constructor & Destructor Documentation

#### 6.21.2.1 FileError() [1/2]

```

PhotoFinish::FileError::FileError (
    const std::string & fp,
    const std::string & m ) [inline]
  
```

Constructor.

#### Parameters

<i>fp</i>	File path
<i>m</i>	Error message

Definition at line 160 of file Exception.hh.

### 6.21.2.2 FileError() [2/2]

```
PhotoFinish::FileError::FileError (
    const std::string & fp ) [inline]
```

Constructor.

Parameters

<i>fp</i>	File path
-----------	-----------

Definition at line 168 of file Exception.hh.

## 6.21.3 Member Function Documentation

### 6.21.3.1 what()

```
virtual const char* PhotoFinish::FileError::what ( ) const [pure virtual], [noexcept]
```

Implements [PhotoFinish::ErrorMsg](#).

Implemented in [PhotoFinish::FileContentError](#), [PhotoFinish::FileOpenError](#), and [PhotoFinish::UnknownFileType](#).

## 6.21.4 Member Data Documentation

### 6.21.4.1 \_filepath

```
const std::string PhotoFinish::FileError::_filepath [protected]
```

Definition at line 152 of file Exception.hh.

The documentation for this class was generated from the following file:

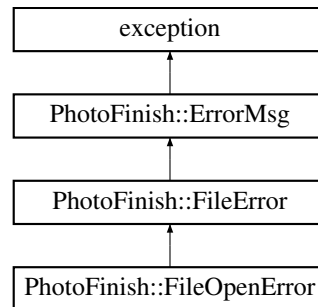
- [Exception.hh](#)

## 6.22 PhotoFinish::FileOpenError Class Reference

File open exception.

```
#include <Exception.hh>
```

Inheritance diagram for PhotoFinish::FileOpenError:



### Public Member Functions

- [FileOpenError](#) (const std::string &fp, const std::string &m)  
*Constructor.*
- [FileOpenError](#) (const std::string &fp)  
*Constructor.*
- virtual const char \* [what](#) () const noexcept

### Additional Inherited Members

#### 6.22.1 Detailed Description

File open exception.

Definition at line 205 of file Exception.hh.

#### 6.22.2 Constructor & Destructor Documentation

##### 6.22.2.1 FileOpenError() [1/2]

```
PhotoFinish::FileOpenError::FileOpenError (
    const std::string & fp,
    const std::string & m ) [inline]
```

Constructor.

**Parameters**

<i>fp</i>	File path
<i>m</i>	Error message

Definition at line 212 of file Exception.hh.

**6.22.2.2 FileOpenError()** [2/2]

```
PhotoFinish::FileOpenError::FileOpenError (
    const std::string & fp ) [inline]
```

Constructor.

**Parameters**

<i>fp</i>	File path
-----------	-----------

Definition at line 220 of file Exception.hh.

**6.22.3 Member Function Documentation****6.22.3.1 what()**

```
virtual const char* PhotoFinish::FileOpenError::what ( ) const [inline], [virtual], [noexcept]
```

Implements [PhotoFinish::FileError](#).

Definition at line 224 of file Exception.hh.

The documentation for this class was generated from the following file:

- [Exception.hh](#)

**6.23 CMS::Format Class Reference**

Wrap LCMS2's pixel format.

```
#include <CMS.hh>
```



## Public Member Functions

- [Format](#) ()  
*Empty constructor.*
- [operator cmsUInt32Number](#) () const  
*Cast to an unsigned int for direct use with LCMS2.*
- [Format](#) & [set\\_8bit](#) (void)  
*Set to 8 bit bytes per channel.*
- bool [is\\_8bit](#) (void) const  
*Is the format 8-bits per channel?*
- [Format](#) & [set\\_16bit](#) (void)  
*Set to 16 bits per channel.*
- bool [is\\_16bit](#) (void) const  
*Is the format 16-bits (integer) per channel?*
- [Format](#) & [set\\_32bit](#) (void)  
*Set to 32 bits per channel.*
- bool [is\\_32bit](#) (void) const  
*Is the format 32-bits (integer) per channel?*
- [Format](#) & [set\\_half](#) (void)  
*Set to 16 bit half-precision floating point values per channel.*
- bool [is\\_half](#) (void) const  
*Is the format a half-precision floating point value(s) per channel?*
- [Format](#) & [set\\_float](#) (void)  
*Set to 32 bit single-precision floating point values per channel.*
- bool [is\\_float](#) (void) const  
*Is the format a single-precision floating point value(s) per channel?*
- [Format](#) & [set\\_double](#) (void)  
*Set to 64 bit double-precision floating point value(s) per channel.*
- bool [is\\_double](#) (void) const  
*Is the format a double-precision floating point value(s) per channel?*
- [Format](#) & [set\\_channel\\_type](#) (unsigned char bytes, bool fp=false)  
*Set the channel type (bytes and float flag)*
- [Format](#) & [set\\_channel\\_type](#) (const [Format](#) &other)  
*Set the channel type (bytes and float flag) from another [Format](#) object.*
- template<typename P >  
[Format](#) & [set\\_channel\\_type](#) (void)  
*Set the channel type (bytes and float flag) from the template type.*
- bool [is\\_integer](#) (void) const  
*Is the format integer?*
- bool [is\\_fp](#) (void) const  
*Is the format floating point?*
- bool [is\\_optimised](#) (void) const
- unsigned int [channels](#) (void) const  
*Get the number of channels.*
- [Format](#) & [set\\_extra\\_channels](#) (unsigned int e)  
*Set the number of 'extra' channels e.g alpha.*
- unsigned int [extra\\_channels](#) (void) const  
*Get the number of 'extra' channels e.g alpha.*
- unsigned int [total\\_channels](#) (void) const  
*Get the total number of channels i.e [channels\(\)](#) + [extra\\_channels\(\)](#)*
- unsigned int [bytes\\_per\\_channel](#) (void) const

- unsigned int [bytes\\_per\\_pixel](#) (void) const
- [Format](#) & [set\\_swap](#) (bool s=true)  
*Set the format as being swapped e.g BGR.*
- [Format](#) & [unset\\_swap](#) (void)  
*Set the format as not being swapped e.g RGB.*
- bool [is\\_swapped](#) (void) const  
*Is the channel order swapped?*
- [Format](#) & [set\\_endianswap](#) (bool e=true)
- [Format](#) & [unset\\_endianswap](#) (void)
- bool [is\\_endianswapped](#) (void) const
- [Format](#) & [set\\_swapfirst](#) (bool f=true)
- [Format](#) & [unset\\_swapfirst](#) (void)
- bool [is\\_swappedfirst](#) (void) const
- [Format](#) & [set\\_planar](#) (bool p=true)  
*Set the format to be planar.*
- [Format](#) & [set\\_packed](#) (void)  
*Set the format to be packed.*
- bool [is\\_planar](#) (void) const  
*Is the format planar?*
- bool [is\\_packed](#) (void) const  
*Is the format packed?*
- [Format](#) & [set\\_vanilla](#) (bool v=true)  
*Set the flavour to 'vanilla' i.e minimum value is white.*
- [Format](#) & [set\\_chocolate](#) (void)  
*Set the flavour to 'chocolate' i.e minimum value is black.*
- bool [is\\_vanilla](#) (void) const  
*Is the flavour 'vanilla'? i.e minimum value is white.*
- bool [is\\_chocolate](#) (void) const  
*Is the flavour 'chocolate'? i.e minimum value is black.*
- [Format](#) & [set\\_colour\\_model](#) (const [ColourModel](#) cm, unsigned int channels=0)
- [ColourModel](#) [colour\\_model](#) (void) const  
*Get the colour model of the pixel format.*
- [Format](#) & [set\\_premult\\_alpha](#) (bool pa=true)
- [Format](#) & [unset\\_premult\\_alpha](#) ()
- bool [is\\_premult\\_alpha](#) (void) const
- template<typename T >  
T [scaleval](#) (void)  
*Get the maximum value used/supported by this format.*
- template<>  
[Format](#) & [set\\_channel\\_type](#) (void)
- template<>  
[Format](#) & [set\\_channel\\_type](#) (void)
- template<>  
[Format](#) & [set\\_channel\\_type](#) (void)
- template<>  
[Format](#) & [set\\_channel\\_type](#) (void)
- template<>  
[Format](#) & [set\\_channel\\_type](#) (void)

## Static Public Member Functions

- static [Format Grey8](#) (void)  
*Named constructor.*
- static [Format Grey16](#) (void)  
*Named constructor.*
- static [Format RGB8](#) (void)  
*Named constructor.*
- static [Format RGB16](#) (void)  
*Named constructor.*
- static [Format CMYK8](#) (void)  
*Named constructor.*
- static [Format LabFloat](#) (void)  
*Named constructor.*
- static [Format LabDouble](#) (void)  
*Named constructor.*

## Friends

- class [Transform](#)

### 6.23.1 Detailed Description

Wrap LCMS2's pixel format.

Definition at line 175 of file CMS.hh.

### 6.23.2 Constructor & Destructor Documentation

#### 6.23.2.1 Format()

```
CMS::Format::Format ( )
```

Empty constructor.

Definition at line 274 of file CMS.cc.

### 6.23.3 Member Function Documentation

#### 6.23.3.1 bytes\_per\_channel()

```
unsigned int CMS::Format::bytes_per_channel (  
    void ) const [inline]
```

Definition at line 279 of file CMS.hh.

#### 6.23.3.2 bytes\_per\_pixel()

```
unsigned int CMS::Format::bytes_per_pixel (  
    void ) const [inline]
```

Definition at line 281 of file CMS.hh.

#### 6.23.3.3 channels()

```
unsigned int CMS::Format::channels (  
    void ) const [inline]
```

Get the number of channels.

Definition at line 268 of file CMS.hh.

#### 6.23.3.4 CMYK8()

```
Format CMS::Format::CMYK8 (  
    void ) [static]
```

Named constructor.

Definition at line 296 of file CMS.cc.

#### 6.23.3.5 colour\_model()

```
ColourModel CMS::Format::colour_model (  
    void ) const [inline]
```

Get the colour model of the pixel format.

Definition at line 334 of file CMS.hh.

#### 6.23.3.6 extra\_channels()

```
unsigned int CMS::Format::extra_channels (
    void ) const [inline]
```

Get the number of 'extra' channels e.g alpha.

Definition at line 274 of file CMS.hh.

#### 6.23.3.7 Grey16()

```
Format CMS::Format::Grey16 (
    void ) [static]
```

Named constructor.

Definition at line 284 of file CMS.cc.

#### 6.23.3.8 Grey8()

```
Format CMS::Format::Grey8 (
    void ) [static]
```

Named constructor.

Definition at line 280 of file CMS.cc.

#### 6.23.3.9 is\_16bit()

```
bool CMS::Format::is_16bit (
    void ) const [inline]
```

Is the format 16-bits (integer) per channel?

Definition at line 223 of file CMS.hh.

#### 6.23.3.10 is\_32bit()

```
bool CMS::Format::is_32bit (
    void ) const [inline]
```

Is the format 32-bits (integer) per channel?

Definition at line 229 of file CMS.hh.

#### 6.23.3.11 is\_8bit()

```
bool CMS::Format::is_8bit (
    void ) const [inline]
```

Is the format 8-bits per channel?

Definition at line 217 of file CMS.hh.

#### 6.23.3.12 is\_chocolate()

```
bool CMS::Format::is_chocolate (
    void ) const [inline]
```

Is the flavour 'chocolate'? i.e minimum value is black.

Definition at line 327 of file CMS.hh.

#### 6.23.3.13 is\_double()

```
bool CMS::Format::is_double (
    void ) const [inline]
```

Is the format a double-precision floating point value(s) per channel?

Definition at line 247 of file CMS.hh.

#### 6.23.3.14 is\_endianswapped()

```
bool CMS::Format::is_endianswapped (
    void ) const [inline]
```

Definition at line 297 of file CMS.hh.

#### 6.23.3.15 is\_float()

```
bool CMS::Format::is_float (
    void ) const [inline]
```

Is the format a single-precision floating point value(s) per channel?

Definition at line 241 of file CMS.hh.

#### 6.23.3.16 is\_fp()

```
bool CMS::Format::is_fp (
    void ) const [inline]
```

Is the format floating point?

Definition at line 263 of file CMS.hh.

#### 6.23.3.17 is\_half()

```
bool CMS::Format::is_half (
    void ) const [inline]
```

Is the format a half-precision floating point value(s) per channel?

Definition at line 235 of file CMS.hh.

#### 6.23.3.18 is\_integer()

```
bool CMS::Format::is_integer (
    void ) const [inline]
```

Is the format integer?

Definition at line 260 of file CMS.hh.

#### 6.23.3.19 is\_optimised()

```
bool CMS::Format::is_optimised (
    void ) const [inline]
```

Definition at line 265 of file CMS.hh.

#### 6.23.3.20 is\_packed()

```
bool CMS::Format::is_packed (
    void ) const [inline]
```

Is the format packed?

Definition at line 315 of file CMS.hh.

**6.23.3.21 is\_planar()**

```
bool CMS::Format::is_planar (
    void ) const [inline]
```

Is the format planar?

Definition at line 312 of file CMS.hh.

**6.23.3.22 is\_premult\_alpha()**

```
bool CMS::Format::is_premult_alpha (
    void ) const [inline]
```

Definition at line 340 of file CMS.hh.

**6.23.3.23 is\_swapped()**

```
bool CMS::Format::is_swapped (
    void ) const [inline]
```

Is the channel order swapped?

Definition at line 290 of file CMS.hh.

**6.23.3.24 is\_swappedfirst()**

```
bool CMS::Format::is_swappedfirst (
    void ) const [inline]
```

Definition at line 303 of file CMS.hh.

**6.23.3.25 is\_vanilla()**

```
bool CMS::Format::is_vanilla (
    void ) const [inline]
```

Is the flavour 'vanilla'? i.e minimum value is white.

Definition at line 324 of file CMS.hh.



#### 6.23.3.26 LabDouble()

```
Format CMS::Format::LabDouble (
    void ) [static]
```

Named constructor.

Definition at line 304 of file CMS.cc.

#### 6.23.3.27 LabFloat()

```
Format CMS::Format::LabFloat (
    void ) [static]
```

Named constructor.

Definition at line 300 of file CMS.cc.

#### 6.23.3.28 operator cmsUInt32Number()

```
CMS::Format::operator cmsUInt32Number ( ) const [inline]
```

Cast to an unsigned int for direct use with LCMS2.

Definition at line 190 of file CMS.hh.

#### 6.23.3.29 RGB16()

```
Format CMS::Format::RGB16 (
    void ) [static]
```

Named constructor.

Definition at line 292 of file CMS.cc.

#### 6.23.3.30 RGB8()

```
Format CMS::Format::RGB8 (
    void ) [static]
```

Named constructor.

Definition at line 288 of file CMS.cc.

#### 6.23.3.31 scaleval()

```
template<typename T >
T CMS::Format::scaleval (
    void ) [inline]
```

Get the maximum value used/supported by this format.

Definition at line 344 of file CMS.hh.

#### 6.23.3.32 set\_16bit()

```
Format & CMS::Format::set_16bit (
    void )
```

Set to 16 bits per channel.

Definition at line 328 of file CMS.cc.

#### 6.23.3.33 set\_32bit()

```
Format & CMS::Format::set_32bit (
    void )
```

Set to 32 bits per channel.

Definition at line 335 of file CMS.cc.

#### 6.23.3.34 set\_8bit()

```
Format & CMS::Format::set_8bit (
    void )
```

Set to 8 bit bytes per channel.

Definition at line 321 of file CMS.cc.

#### 6.23.3.35 set\_channel\_type() [1/8]

```
Format & CMS::Format::set_channel_type (
    unsigned char bytes,
    bool fp = false )
```

Set the channel type (bytes and float flag)

Definition at line 366 of file CMS.cc.

**6.23.3.36** `set_channel_type()` [2/8]

```
Format & CMS::Format::set_channel_type (
    const Format & other )
```

Set the channel type (bytes and float flag) from another [Format](#) object.

Definition at line 374 of file CMS.cc.

**6.23.3.37** `set_channel_type()` [3/8]

```
template<typename P >
Format& CMS::Format::set_channel_type (
    void )
```

Set the channel type (bytes and float flag) from the template type.

**6.23.3.38** `set_channel_type()` [4/8]

```
template<>
Format& CMS::Format::set_channel_type (
    void ) [inline]
```

Definition at line 360 of file CMS.hh.

**6.23.3.39** `set_channel_type()` [5/8]

```
template<>
Format& CMS::Format::set_channel_type (
    void ) [inline]
```

Definition at line 363 of file CMS.hh.

**6.23.3.40** `set_channel_type()` [6/8]

```
template<>
Format& CMS::Format::set_channel_type (
    void ) [inline]
```

Definition at line 366 of file CMS.hh.

**6.23.3.41 set\_channel\_type()** [7/8]

```
template<>
Format& CMS::Format::set_channel_type (
    void ) [inline]
```

Definition at line 369 of file CMS.hh.

**6.23.3.42 set\_channel\_type()** [8/8]

```
template<>
Format& CMS::Format::set_channel_type (
    void ) [inline]
```

Definition at line 372 of file CMS.hh.

**6.23.3.43 set\_chocolate()**

```
Format & CMS::Format::set_chocolate (
    void )
```

Set the flavour to 'chocolate' i.e minimum value is black.

Definition at line 438 of file CMS.cc.

**6.23.3.44 set\_colour\_model()**

```
Format & CMS::Format::set_colour_model (
    const ColourModel cm,
    unsigned int channels = 0 )
```

Set the colour model and number of channels 'channels' is only used if the colour model is unknown

Definition at line 443 of file CMS.cc.

**6.23.3.45 set\_double()**

```
Format & CMS::Format::set_double (
    void )
```

Set to 64 bit double-precision floating point value(s) per channel.

Definition at line 358 of file CMS.cc.

**6.23.3.46 set\_endianswap()**

```
Format & CMS::Format::set_endianswap (
    bool e = true )
```

Definition at line 399 of file CMS.cc.

**6.23.3.47 set\_extra\_channels()**

```
Format & CMS::Format::set_extra_channels (
    unsigned int e )
```

Set the number of 'extra' channels e.g alpha.

Definition at line 382 of file CMS.cc.

**6.23.3.48 set\_float()**

```
Format & CMS::Format::set_float (
    void )
```

Set to 32 bit single-precision floating point values per channel.

Definition at line 350 of file CMS.cc.

**6.23.3.49 set\_half()**

```
Format & CMS::Format::set_half (
    void )
```

Set to 16 bit half-precision floating point values per channel.

Definition at line 342 of file CMS.cc.

**6.23.3.50 set\_packed()**

```
Format & CMS::Format::set_packed (
    void )
```

Set the format to be packed.

Definition at line 427 of file CMS.cc.

#### 6.23.3.51 set\_planar()

```
Format & CMS::Format::set_planar (
    bool p = true )
```

Set the format to be planar.

Definition at line 421 of file CMS.cc.

#### 6.23.3.52 set\_premult\_alpha()

```
Format & CMS::Format::set_premult_alpha (
    bool pa = true )
```

Definition at line 486 of file CMS.cc.

#### 6.23.3.53 set\_swap()

```
Format & CMS::Format::set_swap (
    bool s = true )
```

Set the format as being swapped e.g BGR.

Definition at line 388 of file CMS.cc.

#### 6.23.3.54 set\_swapfirst()

```
Format & CMS::Format::set_swapfirst (
    bool f = true )
```

Definition at line 410 of file CMS.cc.

#### 6.23.3.55 set\_vanilla()

```
Format & CMS::Format::set_vanilla (
    bool v = true )
```

Set the flavour to 'vanilla' i.e minimum value is white.

Definition at line 432 of file CMS.cc.

#### 6.23.3.56 total\_channels()

```
unsigned int CMS::Format::total_channels (
    void ) const [inline]
```

Get the total number of channels i.e [channels\(\)](#) + [extra\\_channels\(\)](#)

Definition at line 277 of file CMS.hh.

#### 6.23.3.57 unset\_endianswap()

```
Format & CMS::Format::unset_endianswap (
    void )
```

Definition at line 405 of file CMS.cc.

#### 6.23.3.58 unset\_premult\_alpha()

```
Format & CMS::Format::unset_premult_alpha ( )
```

Definition at line 491 of file CMS.cc.

#### 6.23.3.59 unset\_swap()

```
Format & CMS::Format::unset_swap (
    void )
```

Set the format as not being swapped e.g RGB.

Definition at line 394 of file CMS.cc.

#### 6.23.3.60 unset\_swapfirst()

```
Format & CMS::Format::unset_swapfirst (
    void )
```

Definition at line 416 of file CMS.cc.

### 6.23.4 Friends And Related Function Documentation

### 6.23.4.1 Transform

```
friend class Transform [friend]
```

Definition at line 183 of file CMS.hh.

The documentation for this class was generated from the following files:

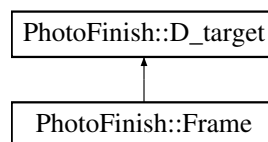
- [CMS.hh](#)
- [CMS.cc](#)

## 6.24 PhotoFinish::Frame Class Reference

Crop+rescaling parameters.

```
#include <Frame.hh>
```

Inheritance diagram for PhotoFinish::Frame:



### Public Types

- typedef std::shared\_ptr< [Frame](#) > ptr  
*Shared pointer for a [Frame](#).*

### Public Member Functions

- [Frame](#) (double tw, double th, double x, double y, double w, double h)  
*Constructor.*
- [Frame](#) (const [D\\_target](#) &target, double x, double y, double w, double h)  
*Constructor.*
- [Image::ptr](#) [crop\\_resize](#) ([Image::ptr](#) img, const [D\\_resize](#) &dr, bool can\_free=false)  
*Crop and resize an image.*
- const double [crop\\_x](#) (void) const  
*The left-most border of the crop window.*
- const double [crop\\_y](#) (void) const  
*The top-most border of the crop window.*
- const double [crop\\_w](#) (void) const  
*The width of the crop window.*
- const double [crop\\_h](#) (void) const  
*The height of the crop window.*
- const double [waste](#) ([Image::ptr](#) img) const  
*How much of the original image is wasted by this crop frame.*



## Additional Inherited Members

### 6.24.1 Detailed Description

Crop+rescaling parameters.

Definition at line 28 of file Frame.hh.

### 6.24.2 Member Typedef Documentation

#### 6.24.2.1 ptr

```
typedef std::shared_ptr<Frame> PhotoFinish::Frame::ptr
```

Shared pointer for a [Frame](#).

Definition at line 71 of file Frame.hh.

### 6.24.3 Constructor & Destructor Documentation

#### 6.24.3.1 Frame() [1/2]

```
PhotoFinish::Frame::Frame (
    double tw,
    double th,
    double x,
    double y,
    double w,
    double h )
```

Constructor.

#### Parameters

<i>tw,th</i>	Size (width, height) of the output
<i>x,y</i>	Top-left corner of crop+rescale window
<i>w,h</i>	Size of the crop+rescale window

Definition at line 28 of file Frame.cc.

### 6.24.3.2 Frame() [2/2]

```
PhotoFinish::Frame::Frame (
    const D_target & target,
    double x,
    double y,
    double w,
    double h )
```

Constructor.

#### Parameters

<i>target</i>	D_target object providing the size (width, height) of the output
<i>x,y</i>	Top-left corner of crop+rescale window
<i>w,h</i>	Size of the crop+rescale window

Definition at line 34 of file Frame.cc.

## 6.24.4 Member Function Documentation

### 6.24.4.1 crop\_h()

```
const double PhotoFinish::Frame::crop_h (
    void ) const [inline]
```

The height of the crop window.

Definition at line 65 of file Frame.hh.

### 6.24.4.2 crop\_resize()

```
Image::ptr PhotoFinish::Frame::crop_resize (
    Image::ptr img,
    const D_resize & dr,
    bool can_free = false )
```

Crop and resize an image.

#### Parameters

<i>img</i>	The source image
<i>dr</i>	A D_resize object which will supply our parameters.
<i>can_free</i>	Can each row of the image be freed after it is convolved?

#### Returns

A new cropped and resized image

Definition at line 40 of file Frame.cc.

##### 6.24.4.3 crop\_w()

```
const double PhotoFinish::Frame::crop_w (
    void ) const [inline]
```

The width of the crop window.

Definition at line 63 of file Frame.hh.

##### 6.24.4.4 crop\_x()

```
const double PhotoFinish::Frame::crop_x (
    void ) const [inline]
```

The left-most border of the crop window.

Definition at line 59 of file Frame.hh.

##### 6.24.4.5 crop\_y()

```
const double PhotoFinish::Frame::crop_y (
    void ) const [inline]
```

The top-most border of the crop window.

Definition at line 61 of file Frame.hh.

##### 6.24.4.6 waste()

```
const double PhotoFinish::Frame::waste (
    Image::ptr img ) const
```

How much of the original image is wasted by this crop frame.

Definition at line 53 of file Frame.cc.

The documentation for this class was generated from the following files:

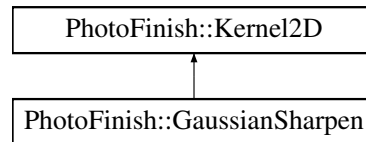
- [Frame.hh](#)
- [Frame.cc](#)

## 6.25 PhotoFinish::GaussianSharpen Class Reference

[GaussianSharpen](#) kernel.

```
#include <Kernel2D.hh>
```

Inheritance diagram for PhotoFinish::GaussianSharpen:



### Public Member Functions

- [GaussianSharpen](#) ()  
*Empty constructor.*
- [GaussianSharpen](#) (const [D\\_sharpen](#) &ds)  
*Constructor.*

### Additional Inherited Members

#### 6.25.1 Detailed Description

[GaussianSharpen](#) kernel.

Definition at line 76 of file Kernel2D.hh.

#### 6.25.2 Constructor & Destructor Documentation

##### 6.25.2.1 [GaussianSharpen\(\)](#) [1/2]

```
PhotoFinish::GaussianSharpen::GaussianSharpen ( )
```

Empty constructor.

Definition at line 262 of file Kernel2D.cc.

##### 6.25.2.2 [GaussianSharpen\(\)](#) [2/2]

```
PhotoFinish::GaussianSharpen::GaussianSharpen (
    const D\_sharpen & ds )
```

Constructor.

## Parameters

<i>ds</i>	A <a href="#">D_sharpen</a> object which will supply our parameters.
-----------	--

Definition at line 265 of file Kernel2D.cc.

The documentation for this class was generated from the following files:

- [Kernel2D.hh](#)
- [Kernel2D.cc](#)

## 6.26 PhotoFinish::Image Class Reference

An image class.

```
#include <Image.hh>
```

### Public Types

- typedef std::shared\_ptr< [Image](#) > [ptr](#)  
*Shared pointer for an [Image](#).*

### Public Member Functions

- [Image](#) (unsigned int w, unsigned int h, [CMS::Format](#) f)  
*Constructor.*
- [~Image](#) ()  
*Destructor.*
- const unsigned int [width](#) (void) const  
*The width of this image.*
- const unsigned int [height](#) (void) const  
*The height of this image.*
- bool [has\\_profile](#) (void) const
- const [CMS::Profile::ptr](#) [profile](#) (void) const  
*Get the ICC profile.*
- void [set\\_profile](#) ([CMS::Profile::ptr](#) p)  
*Set the ICC profile.*
- [CMS::Format](#) [format](#) (void) const  
*Get the [CMS](#) format.*
- const [definable](#)< double > [xres](#) (void) const  
*The X resolution of this image (PPI)*
- const [definable](#)< double > [yres](#) (void) const  
*The Y resolution of this image (PPI)*
- void [set\\_resolution](#) (double r)  
*Set both the X and Y resolution (PPI)*
- void [set\\_xres](#) (double r)  
*Set the X resolution (PPI)*

- void [set\\_yres](#) (double r)  
*Set the Y resolution (PPI)*
- void [set\\_resolution](#) (double xr, double yr)  
*Set the X and Y resolutions (PPI)*
- void [set\\_resolution\\_from\\_size](#) (double size)  
*Set the resolution given the length of the longest side (in inches)*
- size\_t [pixel\\_size](#) (void) const  
*Return the size of a pixel in bytes.*
- size\_t [row\\_size](#) (void) const  
*Return the size of a row in bytes.*
- void [check\\_rowdata\\_alloc](#) (unsigned int y)
- template<typename T = unsigned char>  
T \* [row](#) (unsigned int y) const  
*Pointer to pixel data at start of row.*
- template<typename T >  
T \* [at](#) (unsigned int x, unsigned int y) const  
*Pointer to pixel data at coordinates.*
- template<typename T >  
T & [at](#) (unsigned int x, unsigned int y, unsigned char c) const
- void [free\\_row](#) (unsigned int y)  
*Free the memory storing row 'y'.*
- Exiv2::ExifData & [EXIFtags](#) (void)  
*The Exiv2::ExifData object.*
- Exiv2::IptcData & [IPTCtags](#) (void)  
*The Exiv2::IptcData object.*
- Exiv2::XmpData & [XMPTags](#) (void)  
*The Exiv2::XmpData object.*
- ptr transform\_colour (CMS::Profile::ptr dest\_profile, CMS::Format dest\_format, CMS::Intent intent=CMS::Intent::Perceptual, bool can\_free=false)  
*Transform this image into a different colour space and/or ICC profile, making a new image.*
- void transform\_colour\_inplace (CMS::Profile::ptr dest\_profile, CMS::Format dest\_format, CMS::Intent intent=CMS::Intent::Perceptual)  
*Transform this image in-place into a different colour space and/or ICC profile.*
- void [un\\_alpha\\_mult](#) (void)  
*Un-pre-multiply the colour values with the alpha channel.*
- void [alpha\\_mult](#) (CMS::Format dest\_format)  
*Pre-multiply the colour values with the alpha.*

## Static Public Member Functions

- static CMS::Profile::ptr default\_profile (CMS::ColourModel default\_colourmodel, std::string for\_desc)  
*Create either an sRGB or greyscale profile depending on image format.*
- static CMS::Profile::ptr default\_profile (CMS::Format format, std::string for\_desc)

### 6.26.1 Detailed Description

An image class.

Definition at line 31 of file Image.hh.

## 6.26.2 Member Typedef Documentation

### 6.26.2.1 ptr

```
typedef std::shared_ptr<Image> PhotoFinish::Image::ptr
```

Shared pointer for an [Image](#).

Definition at line 55 of file Image.hh.

## 6.26.3 Constructor & Destructor Documentation

### 6.26.3.1 Image()

```
PhotoFinish::Image::Image (
    unsigned int w,
    unsigned int h,
    CMS::Format f )
```

Constructor.

#### Parameters

<i>w,h</i>	Width and height of the image
<i>t</i>	LCMS2 pixel format

Definition at line 29 of file Image.cc.

### 6.26.3.2 ~Image()

```
PhotoFinish::Image::~Image ( )
```

Destructor.

Definition at line 45 of file Image.cc.

## 6.26.4 Member Function Documentation

#### 6.26.4.1 alpha\_mult()

```
void PhotoFinish::Image::alpha_mult (
    CMS::Format dest_format )
```

Pre-multiply the colour values with the alpha.



## Parameters

<i>dest_format</i>	Destination format, only the channel type (bytes and float flag) are used.
--------------------	--

Definition at line 444 of file Image.cc.

6.26.4.2 `at()` [1/2]

```
template<typename T >
T* PhotoFinish::Image::at (
    unsigned int x,
    unsigned int y ) const [inline]
```

Pointer to pixel data at coordinates.

Definition at line 122 of file Image.hh.

6.26.4.3 `at()` [2/2]

```
template<typename T >
T& PhotoFinish::Image::at (
    unsigned int x,
    unsigned int y,
    unsigned char c ) const [inline]
```

Definition at line 125 of file Image.hh.

6.26.4.4 `check_rowdata_alloc()`

```
void PhotoFinish::Image::check_rowdata_alloc (
    unsigned int y ) [inline]
```

Definition at line 111 of file Image.hh.

6.26.4.5 `default_profile()` [1/2]

```
CMS::Profile::ptr PhotoFinish::Image::default_profile (
    CMS::ColourModel default_colourmodel,
    std::string for_desc ) [static]
```

Create either an sRGB or greyscale profile depending on image format.

Definition at line 57 of file Image.cc.

#### 6.26.4.6 default\_profile() [2/2]

```
static CMS::Profile::ptr PhotoFinish::Image::default_profile (
    CMS::Format format,
    std::string for_desc ) [inline], [static]
```

Definition at line 147 of file Image.hh.

#### 6.26.4.7 EXIFtags()

```
Exiv2::ExifData& PhotoFinish::Image::EXIFtags (
    void ) [inline]
```

The Exiv2::ExifData object.

Definition at line 136 of file Image.hh.

#### 6.26.4.8 format()

```
CMS::Format PhotoFinish::Image::format (
    void ) const [inline]
```

Get the CMS format.

Definition at line 82 of file Image.hh.

#### 6.26.4.9 free\_row()

```
void PhotoFinish::Image::free_row (
    unsigned int y ) [inline]
```

Free the memory storing row 'y'.

Definition at line 128 of file Image.hh.

#### 6.26.4.10 has\_profile()

```
bool PhotoFinish::Image::has_profile (
    void ) const [inline]
```

Definition at line 73 of file Image.hh.

#### 6.26.4.11 height()

```
const unsigned int PhotoFinish::Image::height (
    void ) const [inline]
```

The height of this image.

Definition at line 71 of file Image.hh.

#### 6.26.4.12 IPTCtags()

```
Exiv2::IptcData& PhotoFinish::Image::IPTCtags (
    void ) [inline]
```

The Exiv2::IptcData object.

Definition at line 139 of file Image.hh.

#### 6.26.4.13 pixel\_size()

```
size_t PhotoFinish::Image::pixel_size (
    void ) const [inline]
```

Return the size of a pixel in bytes.

Definition at line 106 of file Image.hh.

#### 6.26.4.14 profile()

```
const CMS::Profile::ptr PhotoFinish::Image::profile (
    void ) const [inline]
```

Get the ICC profile.

Definition at line 76 of file Image.hh.

#### 6.26.4.15 row()

```
template<typename T = unsigned char>
T* PhotoFinish::Image::row (
    unsigned int y ) const [inline]
```

Pointer to pixel data at start of row.

Definition at line 118 of file Image.hh.

#### 6.26.4.16 row\_size()

```
size_t PhotoFinish::Image::row_size (
    void ) const [inline]
```

Return the size of a row in bytes.

Definition at line 109 of file Image.hh.

#### 6.26.4.17 set\_profile()

```
void PhotoFinish::Image::set_profile (
    CMS::Profile::ptr p ) [inline]
```

Set the ICC profile.

Definition at line 79 of file Image.hh.

#### 6.26.4.18 set\_resolution() [1/2]

```
void PhotoFinish::Image::set_resolution (
    double r ) [inline]
```

Set both the X and Y resolution (PPI)

Definition at line 91 of file Image.hh.

#### 6.26.4.19 set\_resolution() [2/2]

```
void PhotoFinish::Image::set_resolution (
    double xr,
    double yr ) [inline]
```

Set the X and Y resolutions (PPI)

Definition at line 100 of file Image.hh.

#### 6.26.4.20 set\_resolution\_from\_size()

```
void PhotoFinish::Image::set_resolution_from_size (
    double size ) [inline]
```

Set the resolution given the length of the longest side (in inches)

Definition at line 103 of file Image.hh.

#### 6.26.4.21 set\_xres()

```
void PhotoFinish::Image::set_xres (
    double r ) [inline]
```

Set the X resolution (PPI)

Definition at line 94 of file Image.hh.

#### 6.26.4.22 set\_yres()

```
void PhotoFinish::Image::set_yres (
    double r ) [inline]
```

Set the Y resolution (PPI)

Definition at line 97 of file Image.hh.

#### 6.26.4.23 transform\_colour()

```
Image::ptr PhotoFinish::Image::transform_colour (
    CMS::Profile::ptr dest_profile,
    CMS::Format dest_format,
    CMS::Intent intent = CMS::Intent::Perceptual,
    bool can_free = false )
```

Transform this image into a different colour space and/or ICC profile, making a new image.

##### Parameters

<i>dest_profile</i>	The ICC profile of the destination. If NULL, uses image's profile.
<i>dest_format</i>	The LCMS2 pixel format.
<i>intent</i>	The ICC intent of the transform, defaults to perceptual.
<i>can_free</i>	Whether rows can be freed after transforming, defaults to false.

##### Returns

A new image

Definition at line 147 of file Image.cc.

#### 6.26.4.24 transform\_colour\_inplace()

```
void PhotoFinish::Image::transform_colour_inplace (
    CMS::Profile::ptr dest_profile,
```

```
CMS::Format dest_format,  
CMS::Intent intent = CMS::Intent::Perceptual )
```

Transform this image in-place into a different colour space and/or ICC profile.

#### Parameters

<i>dest_profile</i>	The ICC profile of the destination. If NULL, uses image's profile.
<i>dest_format</i>	The LCMS2 pixel format.
<i>intent</i>	The ICC intent of the transform, defaults to perceptual.

Definition at line 214 of file Image.cc.

#### 6.26.4.25 un\_alpha\_mult()

```
void PhotoFinish::Image::un_alpha_mult (  
    void )
```

Un-pre-multiply the colour values with the alpha channel.

Converts data to floating point (SAMPLE) in the process

Definition at line 347 of file Image.cc.

#### 6.26.4.26 width()

```
const unsigned int PhotoFinish::Image::width (  
    void ) const [inline]
```

The width of this image.

Definition at line 68 of file Image.hh.

#### 6.26.4.27 XMPTags()

```
Exiv2::XmpData& PhotoFinish::Image::XMPTags (  
    void ) [inline]
```

The Exiv2::XmpData object.

Definition at line 142 of file Image.hh.

## 6.26.4.28 xres()

```
const definable<double> PhotoFinish::Image::xres (
    void ) const [inline]
```

The X resolution of this image (PPI)

Definition at line 85 of file Image.hh.

## 6.26.4.29 yres()

```
const definable<double> PhotoFinish::Image::yres (
    void ) const [inline]
```

The Y resolution of this image (PPI)

Definition at line 88 of file Image.hh.

The documentation for this class was generated from the following files:

- [Image.hh](#)
- [Image.cc](#)

## 6.27 PhotoFinish::ImageFilepath Class Reference

Class for holding filename and the image format.

```
#include <ImageFile.hh>
```

### Public Member Functions

- [ImageFilepath](#) (const fs::path [filepath](#), const std::string [format](#))  
*Constructor.*
- [ImageFilepath](#) (const fs::path [filepath](#))  
*Constructor.*
- fs::path [fixed\\_filepath](#) (void) const
- void [fix\\_filepath](#) (void)
- virtual const fs::path [filepath](#) (void) const  
*File path of this image file.*
- virtual std::string [format](#) (void) const  
*Format of this image file.*

### Friends

- std::ostream & [operator<<](#) (std::ostream &out, const [ImageFilepath](#) &fp)

### 6.27.1 Detailed Description

Class for holding filename and the image format.

Definition at line 55 of file ImageFile.hh.

### 6.27.2 Constructor & Destructor Documentation

#### 6.27.2.1 ImageFilepath() [1/2]

```
PhotoFinish::ImageFilepath::ImageFilepath (
    const fs::path filepath,
    const std::string format )
```

Constructor.

##### Parameters

<i>filepath</i>	The path of the image file
<i>format</i>	Format of the image file

Definition at line 28 of file ImageFile.cc.

#### 6.27.2.2 ImageFilepath() [2/2]

```
PhotoFinish::ImageFilepath::ImageFilepath (
    const fs::path filepath )
```

Constructor.

Guess the format from the file extension.

##### Parameters

<i>filepath</i>	The path of the image file
-----------------	----------------------------

Definition at line 33 of file ImageFile.cc.

### 6.27.3 Member Function Documentation



### 6.27.3.1 filepath()

```
virtual const fs::path PhotoFinish::ImageFilepath::filepath (  
    void ) const [inline], [virtual]
```

File path of this image file.

Definition at line 80 of file ImageFile.hh.

### 6.27.3.2 fix\_filepath()

```
void PhotoFinish::ImageFilepath::fix_filepath (  
    void ) [inline]
```

Definition at line 77 of file ImageFile.hh.

### 6.27.3.3 fixed\_filepath()

```
fs::path PhotoFinish::ImageFilepath::fixed_filepath (  
    void ) const
```

Definition at line 84 of file ImageFile.cc.

### 6.27.3.4 format()

```
virtual std::string PhotoFinish::ImageFilepath::format (  
    void ) const [inline], [virtual]
```

Format of this image file.

Definition at line 83 of file ImageFile.hh.

## 6.27.4 Friends And Related Function Documentation

### 6.27.4.1 operator<<

```
std::ostream& operator<< (  
    std::ostream & out,  
    const ImageFilepath & fp ) [friend]
```

Definition at line 85 of file ImageFile.hh.

The documentation for this class was generated from the following files:

- [ImageFile.hh](#)
- [ImageFile.cc](#)

## 6.28 PhotoFinish::ImageReader Class Reference

Abstract base class for reading image files.

```
#include <ImageFile.hh>
```

### Public Types

- typedef std::shared\_ptr< [ImageReader](#) > [ptr](#)  
*Shared pointer for an [ImageReader](#).*

### Public Member Functions

- virtual [Image::ptr](#) [read](#) (void)  
*Read the file into an image.*
- virtual [Image::ptr](#) [read](#) ([Destination::ptr](#) dest)=0  
*Read the file into an image.*

### Static Public Member Functions

- static [ImageReader::ptr](#) [open](#) (const [ImageFilepath](#) &ifp)  
*Named constructor.*

### Protected Member Functions

- [ImageReader](#) (const fs::path fp)  
*Private constructor.*
- virtual void [extract\\_tags](#) ([Image::ptr](#) img)  
*Extract tags from file.*

### Protected Attributes

- const fs::path [\\_filepath](#)
- bool [\\_is\\_open](#)

#### 6.28.1 Detailed Description

Abstract base class for reading image files.

Definition at line 96 of file ImageFile.hh.

#### 6.28.2 Member Typedef Documentation

### 6.28.2.1 ptr

```
typedef std::shared_ptr<ImageReader> PhotoFinish::ImageReader::ptr
```

Shared pointer for an [ImageReader](#).

Definition at line 109 of file ImageFile.hh.

## 6.28.3 Constructor & Destructor Documentation

### 6.28.3.1 ImageReader()

```
PhotoFinish::ImageReader::ImageReader (
    const fs::path fp ) [protected]
```

Private constructor.

Definition at line 126 of file ImageFile.cc.

## 6.28.4 Member Function Documentation

### 6.28.4.1 extract\_tags()

```
void PhotoFinish::ImageReader::extract_tags (
    Image::ptr img ) [protected], [virtual]
```

Extract tags from file.

Definition at line 131 of file ImageFile.cc.

### 6.28.4.2 open()

```
ImageReader::ptr PhotoFinish::ImageReader::open (
    const ImageFilepath & ifp ) [static]
```

Named constructor.

Use the extension of the file path to decide what class to use

**Parameters**

<i>filepath</i>	File path
-----------------	-----------

Definition at line 149 of file ImageFile.cc.

**6.28.4.3 read()** [1/2]

```
Image::ptr PhotoFinish::ImageReader::read (  
    void ) [virtual]
```

Read the file into an image.

**Returns**

A new [Image](#) object

Definition at line 187 of file ImageFile.cc.

**6.28.4.4 read()** [2/2]

```
virtual Image::ptr PhotoFinish::ImageReader::read (  
    Destination::ptr dest ) [pure virtual]
```

Read the file into an image.

**Parameters**

<i>dest</i>	A <a href="#">Destination</a> object where some information from the file will be placed
-------------	--

**Returns**

A new [Image](#) object

**6.28.5 Member Data Documentation****6.28.5.1 \_filepath**

```
const fs::path PhotoFinish::ImageReader::_filepath [protected]
```

Definition at line 98 of file ImageFile.hh.

6.28.5.2 `_is_open`

```
bool PhotoFinish::ImageReader::_is_open [protected]
```

Definition at line 99 of file ImageFile.hh.

The documentation for this class was generated from the following files:

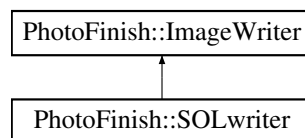
- [ImageFile.hh](#)
- [ImageFile.cc](#)

## 6.29 PhotoFinish::ImageWriter Class Reference

Abstract base class for writing image files.

```
#include <ImageFile.hh>
```

Inheritance diagram for PhotoFinish::ImageWriter:



### Public Types

- typedef std::shared\_ptr< [ImageWriter](#) > ptr  
*Shared pointer for an [ImageWriter](#).*

### Public Member Functions

- virtual [CMS::Format preferred\\_format](#) ([CMS::Format](#) format)=0  
*Modify an LCMS2 pixel format into a "type" that the file format can write.*
- virtual void [write](#) ([Image::ptr](#) img, [Destination::ptr](#) dest, bool can\_free=false)=0  
*Write an image to the file.*

### Static Public Member Functions

- static [ImageWriter::ptr open](#) (const [ImageFilepath](#) &ifp)  
*Named constructor.*
- static void [add\\_variables](#) ([Destination::ptr](#) dest, [multihash](#) &vars)  
*Add variables to one of the configuration objects based on destination format.*

### Protected Member Functions

- [ImageWriter](#) (const fs::path fp)  
*Private constructor.*
- virtual void [embed\\_tags](#) ([Image::ptr](#) img) const

## Protected Attributes

- const fs::path [\\_filepath](#)
- bool [\\_is\\_open](#)

### 6.29.1 Detailed Description

Abstract base class for writing image files.

Definition at line 135 of file ImageFile.hh.

### 6.29.2 Member Typedef Documentation

#### 6.29.2.1 ptr

```
typedef std::shared_ptr<ImageWriter> PhotoFinish::ImageWriter::ptr
```

Shared pointer for an [ImageWriter](#).

Definition at line 147 of file ImageFile.hh.

### 6.29.3 Constructor & Destructor Documentation

#### 6.29.3.1 ImageWriter()

```
PhotoFinish::ImageWriter::ImageWriter (  
    const fs::path fp ) [protected]
```

Private constructor.

Definition at line 193 of file ImageFile.cc.

### 6.29.4 Member Function Documentation

## 6.29.4.1 add\_variables()

```
void PhotoFinish::ImageWriter::add_variables (
    Destination::ptr dest,
    multihash & vars ) [static]
```

Add variables to one of the configuration objects based on destination format.

Definition at line 254 of file ImageFile.cc.

## 6.29.4.2 embed\_tags()

```
void PhotoFinish::ImageWriter::embed_tags (
    Image::ptr img ) const [protected], [virtual]
```

Definition at line 198 of file ImageFile.cc.

## 6.29.4.3 open()

```
ImageWriter::ptr PhotoFinish::ImageWriter::open (
    const ImageFilepath & ifp ) [static]
```

Named constructor.

Use the extension of the file path to decide what class to use

## Parameters

<i>filepath</i>	File path
-----------------	-----------

Definition at line 211 of file ImageFile.cc.

## 6.29.4.4 preferred\_format()

```
virtual CMS::Format PhotoFinish::ImageWriter::preferred_format (
    CMS::Format format ) [pure virtual]
```

Modify an LCMS2 pixel format into a "type" that the file format can write.

Implemented in [PhotoFinish::SOLwriter](#).

#### 6.29.4.5 write()

```
virtual void PhotoFinish::ImageWriter::write (
    Image::ptr img,
    Destination::ptr dest,
    bool can_free = false ) [pure virtual]
```

Write an image to the file.

##### Parameters

<i>img</i>	The <a href="#">Image</a> object to write
<i>dest</i>	A <a href="#">Destination</a> object, used for the JPEG/PNG/etc parameters
<i>can_free</i>	Can each row of the image be freed after it is written?

Implemented in [PhotoFinish::SOLwriter](#).

### 6.29.5 Member Data Documentation

#### 6.29.5.1 \_filepath

```
const fs::path PhotoFinish::ImageWriter::_filepath [protected]
```

Definition at line 137 of file ImageFile.hh.

#### 6.29.5.2 \_is\_open

```
bool PhotoFinish::ImageWriter::_is_open [protected]
```

Definition at line 138 of file ImageFile.hh.

The documentation for this class was generated from the following files:

- [ImageFile.hh](#)
- [ImageFile.cc](#)

### 6.30 PhotoFinish::jpeg\_destination\_state\_t Struct Reference

Structure holding information for the ostream writer.



## Public Attributes

- JOCTET \* [buffer](#)
- std::ostream \* [os](#)
- size\_t [buffer\\_size](#)

### 6.30.1 Detailed Description

Structure holding information for the ostream writer.

Definition at line 106 of file JPEG\_ostream.cc.

### 6.30.2 Member Data Documentation

#### 6.30.2.1 [buffer](#)

JOCTET\* PhotoFinish::jpeg\_destination\_state\_t::buffer

Definition at line 107 of file JPEG\_ostream.cc.

#### 6.30.2.2 [buffer\\_size](#)

size\_t PhotoFinish::jpeg\_destination\_state\_t::buffer\_size

Definition at line 109 of file JPEG\_ostream.cc.

#### 6.30.2.3 [os](#)

std::ostream\* PhotoFinish::jpeg\_destination\_state\_t::os

Definition at line 108 of file JPEG\_ostream.cc.

The documentation for this struct was generated from the following file:

- [JPEG\\_ostream.cc](#)

## 6.31 PhotoFinish::jpeg\_source\_state\_t Struct Reference

Structure holding information for the istream reader.

## Public Attributes

- JOCTET \* [buffer](#)
- std::istream \* [is](#)
- size\_t [buffer\\_size](#)

### 6.31.1 Detailed Description

Structure holding information for the istream reader.

Definition at line 27 of file JPEG\_iostream.cc.

### 6.31.2 Member Data Documentation

#### 6.31.2.1 [buffer](#)

```
JOCTET* PhotoFinish::jpeg_source_state_t::buffer
```

Definition at line 28 of file JPEG\_iostream.cc.

#### 6.31.2.2 [buffer\\_size](#)

```
size_t PhotoFinish::jpeg_source_state_t::buffer_size
```

Definition at line 30 of file JPEG\_iostream.cc.

#### 6.31.2.3 [is](#)

```
std::istream* PhotoFinish::jpeg_source_state_t::is
```

Definition at line 29 of file JPEG\_iostream.cc.

The documentation for this struct was generated from the following file:

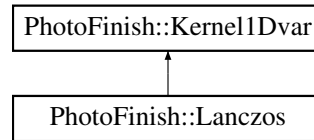
- [JPEG\\_iostream.cc](#)

## 6.32 PhotoFinish::Kernel1Dvar Class Reference

Creates and stores coefficients for cropping and resizing an image.

```
#include <Kernel1Dvar.hh>
```

Inheritance diagram for PhotoFinish::Kernel1Dvar:



### Public Types

- typedef std::shared\_ptr< [Kernel1Dvar](#) > ptr  
*Shared pointer for a [Kernel1Dvar](#).*

### Public Member Functions

- [Kernel1Dvar](#) ()  
*Empty constructor.*
- ~[Kernel1Dvar](#) ()  
*Destructor.*
- [Image::ptr](#) convolve\_h ([Image::ptr](#) img, bool can\_free=false)  
*Convolve an image horizontally with this kernel.*
- [Image::ptr](#) convolve\_v ([Image::ptr](#) img, bool can\_free=false)  
*Convolve an image vertically with this kernel.*

### Static Public Member Functions

- static [ptr](#) create (const [D\\_resize](#) &dr, double from\_start, double from\_size, unsigned int from\_max, double to\_size)  
*Named constructor.*

### Protected Member Functions

- [Kernel1Dvar](#) (double to\_size)  
*Private constructor.*
- void build (double from\_start, double from\_size, unsigned int from\_max)  
*Build the kernel; used by derived classes.*
- virtual double range (void) const =0  
*The size of this filter.*
- virtual [SAMPLE](#) eval (double x) const =0  
*Evaluate the filter at a given point.*
- template<typename T, int channels>  
void convolve\_h\_type\_channels ([Image::ptr](#) src, [Image::ptr](#) dest, bool can\_free=false)
- template<typename T >  
void convolve\_h\_type ([Image::ptr](#) src, [Image::ptr](#) dest, bool can\_free=false)
- template<typename T, int channels>  
void convolve\_v\_type\_channels ([Image::ptr](#) src, [Image::ptr](#) dest, bool can\_free=false)
- template<typename T >  
void convolve\_v\_type ([Image::ptr](#) src, [Image::ptr](#) dest, bool can\_free=false)

## Protected Attributes

- unsigned int \* [\\_size](#)
- unsigned int \* [\\_start](#)
- [SAMPLE](#) \*\* [\\_weights](#)
- double [\\_scale](#)
- double [\\_to\\_size](#)
- unsigned int [\\_to\\_size\\_i](#)

### 6.32.1 Detailed Description

Creates and stores coefficients for cropping and resizing an image.

Definition at line 31 of file Kernel1Dvar.hh.

### 6.32.2 Member Typedef Documentation

#### 6.32.2.1 ptr

```
typedef std::shared_ptr<Kernel1Dvar> PhotoFinish::Kernel1Dvar::ptr
```

Shared pointer for a [Kernel1Dvar](#).

Definition at line 64 of file Kernel1Dvar.hh.

### 6.32.3 Constructor & Destructor Documentation

#### 6.32.3.1 Kernel1Dvar() [1/2]

```
PhotoFinish::Kernel1Dvar::Kernel1Dvar (
    double to_size ) [protected]
```

Private constructor.

Definition at line 38 of file Kernel1Dvar.cc.

#### 6.32.3.2 Kernel1Dvar() [2/2]

```
PhotoFinish::Kernel1Dvar::Kernel1Dvar ( )
```

Emoty constructor.

Definition at line 33 of file Kernel1Dvar.cc.

### 6.32.3.3 ~Kernel1Dvar()

```
PhotoFinish::Kernel1Dvar::~~Kernel1Dvar ( )
```

Destructor.

Definition at line 105 of file Kernel1Dvar.cc.

## 6.32.4 Member Function Documentation

### 6.32.4.1 build()

```
void PhotoFinish::Kernel1Dvar::build (
    double from_start,
    double from_size,
    unsigned int from_max ) [protected]
```

Build the kernel; used by derived classes.

Definition at line 49 of file Kernel1Dvar.cc.

### 6.32.4.2 convolve\_h()

```
Image::ptr PhotoFinish::Kernel1Dvar::convolve_h (
    Image::ptr img,
    bool can_free = false )
```

Convolve an image horizontally with this kernel.

Convolve an image horizontally.

#### Parameters

<i>img</i>	Source image
<i>can_free</i>	Can each row of the image be freed after it is convolved?

#### Returns

New image

Definition at line 248 of file Kernel1Dvar.cc.

#### 6.32.4.3 convolve\_h\_type()

```
template<typename T >
void PhotoFinish::Kernel1Dvar::convolve_h_type (
    Image::ptr src,
    Image::ptr dest,
    bool can_free = false ) [protected]
```

Definition at line 179 of file Kernel1Dvar.cc.

#### 6.32.4.4 convolve\_h\_type\_channels()

```
template<typename T , int channels>
void PhotoFinish::Kernel1Dvar::convolve_h_type_channels (
    Image::ptr src,
    Image::ptr dest,
    bool can_free = false ) [protected]
```

Definition at line 126 of file Kernel1Dvar.cc.

#### 6.32.4.5 convolve\_v()

```
Image::ptr PhotoFinish::Kernel1Dvar::convolve_v (
    Image::ptr img,
    bool can_free = false )
```

Convolve an image vertically with this kernel.

Convolve an image vertically.

##### Parameters

<i>img</i>	Source image
<i>can_free</i>	Can each row of the image be freed after it is convolved?

##### Returns

New image

Definition at line 433 of file Kernel1Dvar.cc.

#### 6.32.4.6 convolve\_v\_type()

```
template<typename T >
void PhotoFinish::Kernel1Dvar::convolve_v_type (
```

```

Image::ptr src,
Image::ptr dest,
bool can_free = false ) [protected]

```

Definition at line 364 of file Kernel1Dvar.cc.

#### 6.32.4.7 convolve\_v\_type\_channels()

```

template<typename T , int channels>
void PhotoFinish::Kernel1Dvar::convolve_v_type_channels (
    Image::ptr src,
    Image::ptr dest,
    bool can_free = false ) [protected]

```

Definition at line 284 of file Kernel1Dvar.cc.

#### 6.32.4.8 create()

```

Kernel1Dvar::ptr PhotoFinish::Kernel1Dvar::create (
    const D_resize & dr,
    double from_start,
    double from_size,
    unsigned int from_max,
    double to_size ) [static]

```

Named constructor.

Create a [Kernel1Dvar](#) object using the filter name in the [D\\_resize](#) object.

##### Parameters

<i>dr</i>	A <a href="#">D_resize</a> object which will supply our parameters.
<i>from_start</i>	The starting point of the crop/resample
<i>from_size</i>	The size of the crop/resample
<i>from_max</i>	The size (maximum dimension) of the input
<i>to_size</i>	The size of the output

Definition at line 89 of file Kernel1Dvar.cc.

#### 6.32.4.9 eval()

```

virtual SAMPLE PhotoFinish::Kernel1Dvar::eval (
    double x ) const [protected], [pure virtual]

```

Evaluate the filter at a given point.

#### 6.32.4.10 range()

```
virtual double PhotoFinish::Kernel1Dvar::range (
    void ) const [protected], [pure virtual]
```

The size of this filter.

### 6.32.5 Member Data Documentation

#### 6.32.5.1 \_scale

```
double PhotoFinish::Kernel1Dvar::_scale [protected]
```

Definition at line 35 of file Kernel1Dvar.hh.

#### 6.32.5.2 \_size

```
unsigned int* PhotoFinish::Kernel1Dvar::_size [protected]
```

Definition at line 33 of file Kernel1Dvar.hh.

#### 6.32.5.3 \_start

```
unsigned int * PhotoFinish::Kernel1Dvar::_start [protected]
```

Definition at line 33 of file Kernel1Dvar.hh.

#### 6.32.5.4 \_to\_size

```
double PhotoFinish::Kernel1Dvar::_to_size [protected]
```

Definition at line 35 of file Kernel1Dvar.hh.

#### 6.32.5.5 \_to\_size\_i

```
unsigned int PhotoFinish::Kernel1Dvar::_to_size_i [protected]
```

Definition at line 36 of file Kernel1Dvar.hh.



6.32.5.6 `_weights`

`SAMPLE** PhotoFinish::Kernel1Dvar::_weights [protected]`

Definition at line 34 of file Kernel1Dvar.hh.

The documentation for this class was generated from the following files:

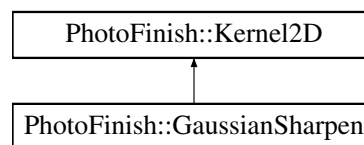
- [Kernel1Dvar.hh](#)
- [Kernel1Dvar.cc](#)

## 6.33 PhotoFinish::Kernel2D Class Reference

Creates and stores coefficients for convolving an image.

```
#include <Kernel2D.hh>
```

Inheritance diagram for PhotoFinish::Kernel2D:



### Public Types

- typedef std::shared\_ptr< [Kernel2D](#) > ptr  
*Shared pointer for a [Kernel2D](#).*

### Public Member Functions

- [Kernel2D](#) ()  
*Empty constructor.*
- [~Kernel2D](#) ()  
*Destructor.*
- [Image::ptr convolve](#) ([Image::ptr](#) img, bool can\_free=false)  
*Convolve and image with this kernel and produce a new image.*

### Static Public Member Functions

- static [ptr create](#) (const [D\\_sharpen](#) &ds)  
*Named constructor.*

## Protected Member Functions

- [Kernel2D](#) (short unsigned int w, short unsigned int h, short unsigned int cx, short unsigned int cy)  
*Private constructor for derived classes.*
- [Kernel2D](#) (short unsigned int size, short unsigned int centre)  
*Private constructor for square filters.*
- `template<typename T >`  
`void convolve\_type (Image::ptr src, Image::ptr dest, bool can_free=false)`
- `template<typename T , int channels>`  
`void convolve\_type\_channels (Image::ptr src, Image::ptr dest, bool can_free=false)`

## Protected Attributes

- short unsigned int [\\_width](#)
- short unsigned int [\\_height](#)
- short unsigned int [\\_centrex](#)
- short unsigned int [\\_centrey](#)
- `SAMPLE ** \_values`

### 6.33.1 Detailed Description

Creates and stores coefficients for convolving an image.

Definition at line 33 of file `Kernel2D.hh`.

### 6.33.2 Member Typedef Documentation

#### 6.33.2.1 ptr

```
typedef std::shared_ptr<Kernel2D> PhotoFinish::Kernel2D::ptr
```

Shared pointer for a [Kernel2D](#).

Definition at line 52 of file `Kernel2D.hh`.

### 6.33.3 Constructor & Destructor Documentation

#### 6.33.3.1 Kernel2D() [1/3]

```
PhotoFinish::Kernel2D::Kernel2D (
    short unsigned int w,
    short unsigned int h,
    short unsigned int cx,
    short unsigned int cy ) [protected]
```

Private constructor for derived classes.

Definition at line 35 of file Kernel2D.cc.

#### 6.33.3.2 Kernel2D() [2/3]

```
PhotoFinish::Kernel2D::Kernel2D (
    short unsigned int size,
    short unsigned int centre ) [protected]
```

Private constructor for square filters.

Definition at line 45 of file Kernel2D.cc.

#### 6.33.3.3 Kernel2D() [3/3]

```
PhotoFinish::Kernel2D::Kernel2D ( )
```

Empty constructor.

Definition at line 29 of file Kernel2D.cc.

#### 6.33.3.4 ~Kernel2D()

```
PhotoFinish::Kernel2D::~~Kernel2D ( )
```

Destructor.

Definition at line 59 of file Kernel2D.cc.

### 6.33.4 Member Function Documentation

#### 6.33.4.1 convolve()

```
Image::ptr PhotoFinish::Kernel2D::convolve (
    Image::ptr img,
    bool can_free = false )
```

Convolve and image with this kernel and produce a new image.

## Parameters

<i>img</i>	Source image
<i>can_free</i>	Can each row of the image be freed after it is convolved?

## Returns

New image

Definition at line 215 of file Kernel2D.cc.

#### 6.33.4.2 convolve\_type()

```
template<typename T >
void PhotoFinish::Kernel2D::convolve_type (
    Image::ptr src,
    Image::ptr dest,
    bool can_free = false ) [protected]
```

Definition at line 147 of file Kernel2D.cc.

#### 6.33.4.3 convolve\_type\_channels()

```
template<typename T , int channels>
void PhotoFinish::Kernel2D::convolve_type_channels (
    Image::ptr src,
    Image::ptr dest,
    bool can_free = false ) [protected]
```

Definition at line 69 of file Kernel2D.cc.

#### 6.33.4.4 create()

```
Kernel2D::ptr PhotoFinish::Kernel2D::create (
    const D_sharpen & ds ) [static]
```

Named constructor.

Create a [Kernel2D](#) object using the parameters in the [D\\_sharpen](#) object.

## Parameters

<i>ds</i>	A <a href="#">D_sharpen</a> object which will supply our parameters.
-----------	--

Definition at line 55 of file Kernel2D.cc.

### 6.33.5 Member Data Documentation

#### 6.33.5.1 \_centrex

```
short unsigned int PhotoFinish::Kernel2D::_centrex [protected]
```

Definition at line 35 of file Kernel2D.hh.

#### 6.33.5.2 \_centrey

```
short unsigned int PhotoFinish::Kernel2D::_centrey [protected]
```

Definition at line 35 of file Kernel2D.hh.

#### 6.33.5.3 \_height

```
short unsigned int PhotoFinish::Kernel2D::_height [protected]
```

Definition at line 35 of file Kernel2D.hh.

#### 6.33.5.4 \_values

```
SAMPLE** PhotoFinish::Kernel2D::_values [protected]
```

Definition at line 36 of file Kernel2D.hh.

#### 6.33.5.5 \_width

```
short unsigned int PhotoFinish::Kernel2D::_width [protected]
```

Definition at line 35 of file Kernel2D.hh.

The documentation for this class was generated from the following files:

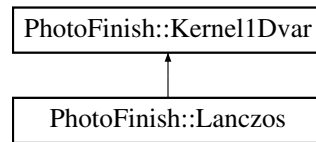
- [Kernel2D.hh](#)
- [Kernel2D.cc](#)

## 6.34 PhotoFinish::Lanczos Class Reference

[Lanczos](#) filter.

```
#include <Kernel1Dvar.hh>
```

Inheritance diagram for PhotoFinish::Lanczos:



### Public Member Functions

- [Lanczos](#) ()  
*Empty constructor.*
- [Lanczos](#) (const [D\\_resize](#) &dr, double from\_start, double from\_size, unsigned int from\_max, double to\_size)  
*Constructor.*

### Additional Inherited Members

#### 6.34.1 Detailed Description

[Lanczos](#) filter.

Definition at line 101 of file Kernel1Dvar.hh.

#### 6.34.2 Constructor & Destructor Documentation

##### 6.34.2.1 [Lanczos](#)() [1/2]

```
PhotoFinish::Lanczos::Lanczos ( )
```

Empty constructor.

Definition at line 469 of file Kernel1Dvar.cc.

##### 6.34.2.2 [Lanczos](#)() [2/2]

```
PhotoFinish::Lanczos::Lanczos (
    const D\_resize & dr,
    double from_start,
    double from_size,
    unsigned int from_max,
    double to_size )
```

Constructor.

## Parameters

<i>dr</i>	A <a href="#">D_resize</a> object which will supply our parameters.
<i>horiz</i>	Will the kernel run in horizontal (true) or vertical direction?
<i>from_start</i>	The starting point of the crop/resample
<i>from_size</i>	The size of the crop/resample
<i>from_max</i>	The size (maximum dimension) of the input
<i>to_size</i>	The size of the output

Definition at line 473 of file Kernel1Dvar.cc.

The documentation for this class was generated from the following files:

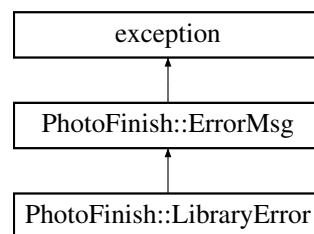
- [Kernel1Dvar.hh](#)
- [Kernel1Dvar.cc](#)

## 6.35 PhotoFinish::LibraryError Class Reference

Library exception.

```
#include <Exception.hh>
```

Inheritance diagram for PhotoFinish::LibraryError:



### Public Member Functions

- [LibraryError](#) (const std::string &l, const std::string &m)  
*Constructor.*
- virtual const char \* [what](#) () const noexcept

### Additional Inherited Members

#### 6.35.1 Detailed Description

Library exception.

Definition at line 283 of file Exception.hh.

## 6.35.2 Constructor & Destructor Documentation

### 6.35.2.1 LibraryError()

```
PhotoFinish::LibraryError::LibraryError (
    const std::string & l,
    const std::string & m ) [inline]
```

Constructor.

#### Parameters

<i>l</i>	Library name
<i>m</i>	Error message

Definition at line 293 of file Exception.hh.

## 6.35.3 Member Function Documentation

### 6.35.3.1 what()

```
virtual const char* PhotoFinish::LibraryError::what ( ) const [inline], [virtual], [noexcept]
```

Implements [PhotoFinish::ErrorMsg](#).

Definition at line 297 of file Exception.hh.

The documentation for this class was generated from the following file:

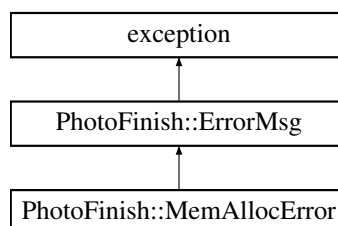
- [Exception.hh](#)

## 6.36 PhotoFinish::MemAllocError Class Reference

Memory allocation exception.

```
#include <Exception.hh>
```

Inheritance diagram for PhotoFinish::MemAllocError:





## Public Member Functions

- [MemAllocError](#) (const std::string &m)  
*Constructor.*
- const char \* [what](#) () const noexcept

## Additional Inherited Members

### 6.36.1 Detailed Description

Memory allocation exception.

Definition at line 134 of file Exception.hh.

### 6.36.2 Constructor & Destructor Documentation

#### 6.36.2.1 MemAllocError()

```
PhotoFinish::MemAllocError::MemAllocError (  
    const std::string & m ) [inline]
```

Constructor.

Parameters

<i>m</i>	Error message
----------	---------------

Definition at line 140 of file Exception.hh.

### 6.36.3 Member Function Documentation

#### 6.36.3.1 what()

```
const char* PhotoFinish::MemAllocError::what ( ) const [inline], [virtual], [noexcept]
```

Implements [PhotoFinish::ErrorMsg](#).

Definition at line 144 of file Exception.hh.

The documentation for this class was generated from the following file:

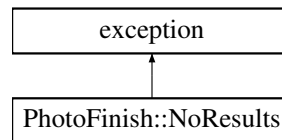
- [Exception.hh](#)

## 6.37 PhotoFinish::NoResults Class Reference

No results exception.

```
#include <Exception.hh>
```

Inheritance diagram for PhotoFinish::NoResults:



### Public Member Functions

- [NoResults](#) (const std::string &c, const std::string &m)  
*Constructor.*
- virtual const char \* [what](#) () const noexcept

### Protected Attributes

- const std::string [\\_class](#)
- const std::string [\\_method](#)

### 6.37.1 Detailed Description

No results exception.

Definition at line 78 of file Exception.hh.

### 6.37.2 Constructor & Destructor Documentation

#### 6.37.2.1 NoResults()

```
PhotoFinish::NoResults::NoResults (
    const std::string & c,
    const std::string & m ) [inline]
```

Constructor.

#### Parameters

<i>c</i>	Class name
<i>m</i>	Method name

Definition at line 88 of file Exception.hh.

### 6.37.3 Member Function Documentation

#### 6.37.3.1 what()

```
virtual const char* PhotoFinish::NoResults::what ( ) const [inline], [virtual], [noexcept]
```

Definition at line 92 of file Exception.hh.

### 6.37.4 Member Data Documentation

#### 6.37.4.1 \_class

```
const std::string PhotoFinish::NoResults::_class [protected]
```

Definition at line 80 of file Exception.hh.

#### 6.37.4.2 \_method

```
const std::string PhotoFinish::NoResults::_method [protected]
```

Definition at line 80 of file Exception.hh.

The documentation for this class was generated from the following file:

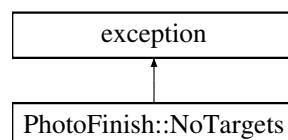
- [Exception.hh](#)

## 6.38 PhotoFinish::NoTargets Class Reference

No targets exception.

```
#include <Exception.hh>
```

Inheritance diagram for PhotoFinish::NoTargets:



## Public Member Functions

- [NoTargets](#) (const std::string &d)  
*Constructor.*
- virtual const char \* [what](#) () const noexcept

## Protected Attributes

- const std::string [\\_destination](#)

### 6.38.1 Detailed Description

No targets exception.

Definition at line 98 of file Exception.hh.

### 6.38.2 Constructor & Destructor Documentation

#### 6.38.2.1 NoTargets()

```
PhotoFinish::NoTargets::NoTargets (
    const std::string & d ) [inline]
```

Constructor.

#### Parameters

<i>d</i>	Name of destination that has no targets
----------	---

Definition at line 107 of file Exception.hh.

### 6.38.3 Member Function Documentation

#### 6.38.3.1 what()

```
virtual const char* PhotoFinish::NoTargets::what ( ) const [inline], [virtual], [noexcept]
```

Definition at line 111 of file Exception.hh.

## 6.38.4 Member Data Documentation

### 6.38.4.1 \_destination

```
const std::string PhotoFinish::NoTargets::_destination [protected]
```

Definition at line 100 of file Exception.hh.

The documentation for this class was generated from the following file:

- [Exception.hh](#)

## 6.39 PhotoFinish::PNGreader\_cb Struct Reference

```
#include <PNGreader_cb.hh>
```

### Public Member Functions

- [PNGreader\\_cb](#) ([Destination::ptr](#) d)
- void [info](#) (png\_structp png, png\_infop info)
- void [row](#) (png\_structp png, png\_bytep row\_data, png\_uint\_32 row\_num, int pass)
- void [end](#) (png\_structp png, png\_infop [info](#))

### Public Attributes

- [Destination::ptr](#) \_destination
- [Image::ptr](#) \_image

### 6.39.1 Detailed Description

Definition at line 28 of file PNGreader\_cb.hh.

### 6.39.2 Constructor & Destructor Documentation

#### 6.39.2.1 PNGreader\_cb()

```
PhotoFinish::PNGreader_cb::PNGreader_cb (
    Destination::ptr d )
```

Definition at line 23 of file PNGreader\_cb.cc.

### 6.39.3 Member Function Documentation

#### 6.39.3.1 end()

```
void PhotoFinish::PNGreader_cb::end (
    png_structp png,
    png_infop info )
```

Definition at line 137 of file PNGreader\_cb.cc.

#### 6.39.3.2 info()

```
void PhotoFinish::PNGreader_cb::info (
    png_structp png,
    png_infop info )
```

Definition at line 27 of file PNGreader\_cb.cc.

#### 6.39.3.3 row()

```
void PhotoFinish::PNGreader_cb::row (
    png_structp png,
    png_bytep row_data,
    png_uint_32 row_num,
    int pass )
```

Definition at line 126 of file PNGreader\_cb.cc.

### 6.39.4 Member Data Documentation

#### 6.39.4.1 \_destination

```
Destination::ptr PhotoFinish::PNGreader_cb::_destination
```

Definition at line 29 of file PNGreader\_cb.hh.

## 6.39.4.2 \_image

`Image::ptr` PhotoFinish::PNGreader\_cb::\_image

Definition at line 30 of file PNGreader\_cb.hh.

The documentation for this struct was generated from the following files:

- [PNGreader\\_cb.hh](#)
- [PNGreader\\_cb.cc](#)

## 6.40 CMS::Profile Class Reference

Wrap LCMS2's cmsHPROFILE.

```
#include <CMS.hh>
```

### Public Types

- `typedef std::shared_ptr< Profile > ptr`  
*Shared pointer typedef.*

### Public Member Functions

- [Profile](#) ()  
*Empty constructor.*
- [Profile](#) (const [Profile](#) &other)  
*Copy constructor.*
- [Profile](#) (fs::path filepath)  
*Constructor from file path.*
- [Profile](#) (const unsigned char \*data, cmsUInt32Number size)  
*Constructor from memory.*
- [Profile](#) (std::istream stream)  
*Constructor from an istream.*
- [~Profile](#) ()  
*Destructor.*
- [operator cmsHPROFILE](#) () const  
*Cast to a profile handle for direct use with LCMS2.*
- void [set\\_description](#) (std::string language, std::string country, std::string text)  
*Set the description tag.*
- void [set\\_description](#) (std::string language, std::string country, std::wstring text)  
*Set the description tag with a wide string.*
- std::string [description](#) (std::string language, std::string country) const  
*Get the description tag.*
- std::wstring [description\\_wide](#) (std::string language, std::string country) const  
*Get the description tag in a wide string.*
- void [set\\_manufacturer](#) (std::string language, std::string country, std::string text)  
*Set the manufacturer tag.*

- void [set\\_manufacturer](#) (std::string language, std::string country, std::wstring text)  
*Set the manufacturer tag with a wide string.*
- std::string [manufacturer](#) (std::string language, std::string country) const  
*Get the manufacturer tag.*
- std::wstring [manufacturer\\_wide](#) (std::string language, std::string country) const  
*Get the manufacturer tag in a wide string.*
- void [set\\_model](#) (std::string language, std::string country, std::string text)  
*Set the model tag.*
- void [set\\_model](#) (std::string language, std::string country, std::wstring text)  
*Set the model tag with a wide string.*
- std::string [model](#) (std::string language, std::string country) const  
*Get the model tag.*
- std::wstring [model\\_wide](#) (std::string language, std::string country) const  
*Get the model tag in a wide string.*
- void [set\\_copyright](#) (std::string language, std::string country, std::string text)  
*Set the copyright tag.*
- void [set\\_copyright](#) (std::string language, std::string country, std::wstring text)  
*Set the copyright tag with a wide string.*
- std::string [copyright](#) (std::string language, std::string country) const  
*Get the copyright tag.*
- std::wstring [copyright\\_wide](#) (std::string language, std::string country) const  
*Get the copyright tag in a wide string.*
- void [save\\_to\\_mem](#) (unsigned char \*&dest, unsigned int &size) const

### Static Public Member Functions

- static [ptr Lab4](#) (void)  
*Named constructor.*
- static [ptr sRGB](#) (void)  
*Named constructor.*
- static [ptr sGrey](#) (void)  
*Named constructor.*

### Friends

- class [\\_\\_gnu\\_cxx::new\\_allocator< Profile >](#)

### 6.40.1 Detailed Description

Wrap LCMS2's cmsHPROFILE.

Definition at line 37 of file CMS.hh.

### 6.40.2 Member Typedef Documentation



### 6.40.2.1 ptr

```
typedef std::shared_ptr<Profile> CMS::Profile::ptr
```

Shared pointer typedef.

Definition at line 81 of file CMS.hh.

## 6.40.3 Constructor & Destructor Documentation

### 6.40.3.1 Profile() [1/5]

```
CMS::Profile::Profile ( )
```

Empty constructor.

Definition at line 34 of file CMS.cc.

### 6.40.3.2 Profile() [2/5]

```
CMS::Profile::Profile (
    const Profile & other )
```

Copy constructor.

Definition at line 39 of file CMS.cc.

### 6.40.3.3 Profile() [3/5]

```
CMS::Profile::Profile (
    fs::path filepath )
```

Constructor from file path.

Definition at line 52 of file CMS.cc.

#### 6.40.3.4 Profile() [4/5]

```
CMS::Profile::Profile (
    const unsigned char * data,
    cmsUInt32Number size )
```

Constructor from memory.

Definition at line 57 of file CMS.cc.

#### 6.40.3.5 Profile() [5/5]

```
CMS::Profile::Profile (
    std::istream stream )
```

Constructor from an istream.

Definition at line 62 of file CMS.cc.

#### 6.40.3.6 ~Profile()

```
CMS::Profile::~~Profile ( )
```

Deconstructor.

Definition at line 67 of file CMS.cc.

### 6.40.4 Member Function Documentation

#### 6.40.4.1 copyright()

```
std::string CMS::Profile::copyright (
    std::string language,
    std::string country ) const
```

Get the copyright tag.

Definition at line 205 of file CMS.cc.

#### 6.40.4.2 copyright\_wide()

```
std::wstring CMS::Profile::copyright_wide (
    std::string language,
    std::string country ) const
```

Get the copyright tag in a wide string.

Definition at line 209 of file CMS.cc.

#### 6.40.4.3 description()

```
std::string CMS::Profile::description (
    std::string language,
    std::string country ) const
```

Get the description tag.

Definition at line 157 of file CMS.cc.

#### 6.40.4.4 description\_wide()

```
std::wstring CMS::Profile::description_wide (
    std::string language,
    std::string country ) const
```

Get the description tag in a wide string.

Definition at line 161 of file CMS.cc.

#### 6.40.4.5 Lab4()

```
Profile::ptr CMS::Profile::Lab4 (
    void ) [static]
```

Named constructor.

Definition at line 72 of file CMS.cc.

#### 6.40.4.6 manufacturer()

```
std::string CMS::Profile::manufacturer (
    std::string language,
    std::string country ) const
```

Get the manufacturer tag.

Definition at line 173 of file CMS.cc.

#### 6.40.4.7 manufacturer\_wide()

```
std::wstring CMS::Profile::manufacturer_wide (
    std::string language,
    std::string country ) const
```

Get the manufacturer tag in a wide string.

Definition at line 177 of file CMS.cc.

#### 6.40.4.8 model()

```
std::string CMS::Profile::model (
    std::string language,
    std::string country ) const
```

Get the model tag.

Definition at line 189 of file CMS.cc.

#### 6.40.4.9 model\_wide()

```
std::wstring CMS::Profile::model_wide (
    std::string language,
    std::string country ) const
```

Get the model tag in a wide string.

Definition at line 193 of file CMS.cc.

#### 6.40.4.10 operator cmsHPROFILE()

```
CMS::Profile::operator cmsHPROFILE ( ) const [inline]
```

Cast to a profile handle for direct use with LCMS2.

Definition at line 78 of file CMS.hh.

#### 6.40.4.11 save\_to\_mem()

```
void CMS::Profile::save_to_mem (
    unsigned char *& dest,
    unsigned int & size ) const
```

Definition at line 213 of file CMS.cc.

#### 6.40.4.12 set\_copyright() [1/2]

```
void CMS::Profile::set_copyright (
    std::string language,
    std::string country,
    std::string text )
```

Set the copyright tag.

Definition at line 197 of file CMS.cc.

#### 6.40.4.13 set\_copyright() [2/2]

```
void CMS::Profile::set_copyright (
    std::string language,
    std::string country,
    std::wstring text )
```

Set the copyright tag with a wide string.

Definition at line 201 of file CMS.cc.

**6.40.4.14** `set_description()` [1/2]

```
void CMS::Profile::set_description (
    std::string language,
    std::string country,
    std::string text )
```

Set the description tag.

Definition at line 149 of file CMS.cc.

**6.40.4.15** `set_description()` [2/2]

```
void CMS::Profile::set_description (
    std::string language,
    std::string country,
    std::wstring text )
```

Set the description tag with a wide string.

Definition at line 153 of file CMS.cc.

**6.40.4.16** `set_manufacturer()` [1/2]

```
void CMS::Profile::set_manufacturer (
    std::string language,
    std::string country,
    std::string text )
```

Set the manufacturer tag.

Definition at line 165 of file CMS.cc.

**6.40.4.17** `set_manufacturer()` [2/2]

```
void CMS::Profile::set_manufacturer (
    std::string language,
    std::string country,
    std::wstring text )
```

Set the manufacturer tag with a wide string.

Definition at line 169 of file CMS.cc.

**6.40.4.18** `set_model()` [1/2]

```
void CMS::Profile::set_model (
    std::string language,
    std::string country,
    std::string text )
```

Set the model tag.

Definition at line 181 of file CMS.cc.

**6.40.4.19** `set_model()` [2/2]

```
void CMS::Profile::set_model (
    std::string language,
    std::string country,
    std::wstring text )
```

Set the model tag with a wide string.

Definition at line 185 of file CMS.cc.

**6.40.4.20** `sGrey()`

```
Profile::ptr CMS::Profile::sGrey (
    void ) [static]
```

Named constructor.

Definition at line 80 of file CMS.cc.

**6.40.4.21** `sRGB()`

```
Profile::ptr CMS::Profile::sRGB (
    void ) [static]
```

Named constructor.

Definition at line 76 of file CMS.cc.

**6.40.5** Friends And Related Function Documentation

#### 6.40.5.1 `__gnu_cxx::new_allocator< Profile >`

```
friend class __gnu_cxx::new_allocator< Profile > [friend]
```

Definition at line 56 of file CMS.hh.

The documentation for this class was generated from the following files:

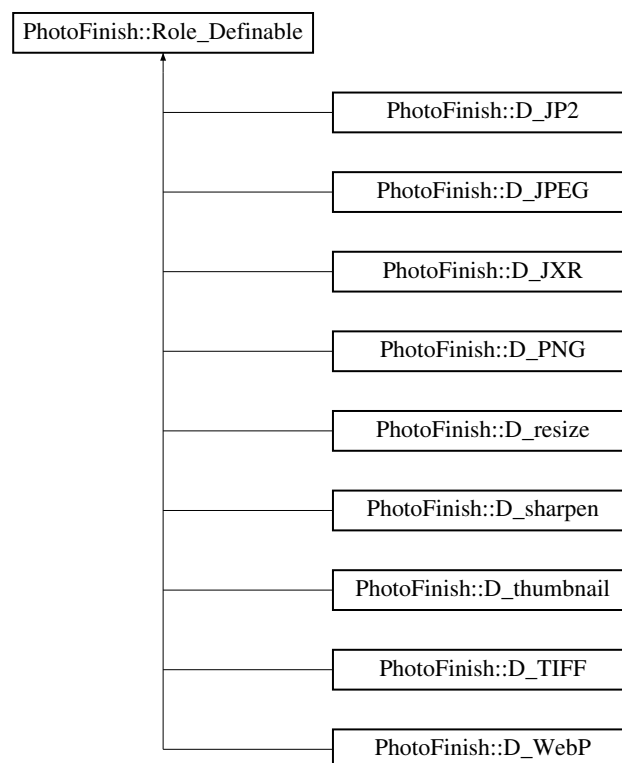
- [CMS.hh](#)
- [CMS.cc](#)

## 6.41 PhotoFinish::Role\_Definable Class Reference

Base class for adding "definable" attribute.

```
#include <Definable.hh>
```

Inheritance diagram for PhotoFinish::Role\_Definable:



### Public Member Functions

- [Role\\_Definable](#) ()  
*Empty constructor.*
- `const bool` [defined](#) (`void`) `const`  
*Is this object defined?*



## Protected Member Functions

- void `set_defined` (bool v=true)  
*Set this object as 'defined' (or not)*
- void `undefine` (void)  
*Undefine the object.*

## Protected Attributes

- bool `_defined`

## Friends

- bool `defined` (const `Role_Definable` &obj)

### 6.41.1 Detailed Description

Base class for adding "definable" attribute.

Definition at line 99 of file Definable.hh.

### 6.41.2 Constructor & Destructor Documentation

#### 6.41.2.1 Role\_Definable()

```
PhotoFinish::Role_Definable::Role_Definable ( ) [inline]
```

Empty constructor.

Sets defined to false

Definition at line 114 of file Definable.hh.

### 6.41.3 Member Function Documentation

#### 6.41.3.1 defined()

```
const bool PhotoFinish::Role_Definable::defined (
    void ) const [inline]
```

Is this object defined?

Definition at line 119 of file Definable.hh.

#### 6.41.3.2 set\_defined()

```
void PhotoFinish::Role_Definable::set_defined (
    bool v = true ) [inline], [protected]
```

Set this object as 'defined' (or not)

Definition at line 104 of file Definable.hh.

#### 6.41.3.3 undefine()

```
void PhotoFinish::Role_Definable::undefine (
    void ) [inline], [protected]
```

Undefine the object.

Definition at line 107 of file Definable.hh.

### 6.41.4 Friends And Related Function Documentation

#### 6.41.4.1 defined

```
bool defined (
    const Role_Definable & obj ) [friend]
```

Definition at line 121 of file Definable.hh.

### 6.41.5 Member Data Documentation

#### 6.41.5.1 \_defined

```
bool PhotoFinish::Role_Definable::_defined [protected]
```

Definition at line 101 of file Definable.hh.

The documentation for this class was generated from the following file:

- [Definable.hh](#)

## 6.42 PhotoFinish::SOLwriter Class Reference

Write the boot logo files for use on Motorola Atrix 4G and possibly other phones.

```
#include <ImageFile.hh>
```

Inheritance diagram for PhotoFinish::SOLwriter:



### Public Member Functions

- [SOLwriter](#) (const fs::path filepath)
- [CMS::Format preferred\\_format](#) ([CMS::Format](#) format)  
*Modify an LCMS2 pixel format into a "type" that the file format can write.*
- void [write](#) ([Image::ptr](#) img, [Destination::ptr](#) dest, bool can\_free=false)  
*Write an image to the file.*

### Additional Inherited Members

#### 6.42.1 Detailed Description

Write the boot logo files for use on Motorola Atrix 4G and possibly other phones.

I haven't been able to find any documentation about this format. It starts with the ASCII string "SOL:" followed by eight null bytes. Then comes the width and height as big-endian 32-bit values. The image data is as uncompressed 5-6-5 bit pixels i.e 16 bits per pixel. No footer.

Definition at line 336 of file ImageFile.hh.

#### 6.42.2 Constructor & Destructor Documentation

##### 6.42.2.1 SOLwriter()

```
PhotoFinish::SOLwriter::SOLwriter (
    const fs::path filepath )
```

Definition at line 26 of file SOLwriter.cc.

#### 6.42.3 Member Function Documentation

### 6.42.3.1 preferred\_format()

```
CMS::Format PhotoFinish::SOLwriter::preferred_format (
    CMS::Format format ) [virtual]
```

Modify an LCMS2 pixel format into a "type" that the file format can write.

Implements [PhotoFinish::ImageWriter](#).

Definition at line 30 of file SOLwriter.cc.

### 6.42.3.2 write()

```
void PhotoFinish::SOLwriter::write (
    Image::ptr img,
    Destination::ptr dest,
    bool can_free = false ) [virtual]
```

Write an image to the file.

#### Parameters

<i>img</i>	The <a href="#">Image</a> object to write
<i>dest</i>	A <a href="#">Destination</a> object, used for the JPEG/PNG/etc parameters
<i>can_free</i>	Can each row of the image be freed after it is written?

Implements [PhotoFinish::ImageWriter](#).

Definition at line 53 of file SOLwriter.cc.

The documentation for this class was generated from the following files:

- [ImageFile.hh](#)
- [SOLwriter.cc](#)

## 6.43 PhotoFinish::Tags Class Reference

Reads and holds tag information.

```
#include <Tags.hh>
```

### Public Types

- typedef std::shared\_ptr< [Tags](#) > ptr  
Shared pointer for a [Tags](#) object.

## Public Member Functions

- [Tags](#) ()  
*Empty Constructor.*
- [Tags](#) (const [Tags](#) &other)  
*Copy constructor.*
- [Tags](#) (const fs::path &filepath)  
*Constructor with a filepath from which to load tags (calls Load)*
- [ptr dupe](#) (void) const  
*Duplicate the tags.*
- void [add\\_searchpath](#) (fs::path path)
- [multihash](#) & [variables](#) (void)  
*The map of variables.*
- Exiv2::ExifData & [EXIFtags](#) (void)  
*The Exiv2::ExifData object.*
- Exiv2::IptcData & [IPTCtags](#) (void)  
*The Exiv2::IptcData object.*
- Exiv2::XmpData & [XMPTags](#) (void)  
*The Exiv2::XmpData object.*
- bool [try\\_load](#) (fs::path filepath)
- void [load](#) (fs::path filepath)  
*Load tags from supplied file path.*
- void [copy\\_from](#) ([Image::ptr](#) img)  
*Copy EXIF/IPTC/XMP tags from an image.*
- void [make\\_thumbnail](#) ([Image::ptr](#) img, const [D\\_thumbnail](#) &dt)  
*Create a thumbnail from the supplied image.*
- void [add\\_resolution](#) ([Image::ptr](#) img)
- void [copy\\_to](#) ([Image::ptr](#) img) const  
*Copy EXIF/IPTC/XMP tags to an image.*

### 6.43.1 Detailed Description

Reads and holds tag information.

Definition at line 41 of file Tags.hh.

### 6.43.2 Member Typedef Documentation

#### 6.43.2.1 ptr

```
typedef std::shared_ptr<Tags> PhotoFinish::Tags::ptr
```

Shared pointer for a [Tags](#) object.

Definition at line 60 of file Tags.hh.

### 6.43.3 Constructor & Destructor Documentation

#### 6.43.3.1 Tags() [1/3]

```
PhotoFinish::Tags::Tags ( )
```

Empty Constructor.

Definition at line 33 of file Tags.cc.

#### 6.43.3.2 Tags() [2/3]

```
PhotoFinish::Tags::Tags (
    const Tags & other )
```

Copy constructor.

Definition at line 36 of file Tags.cc.

#### 6.43.3.3 Tags() [3/3]

```
PhotoFinish::Tags::Tags (
    const fs::path & filepath )
```

Constructor with a filepath from which to load tags (calls Load)

Definition at line 44 of file Tags.cc.

### 6.43.4 Member Function Documentation

#### 6.43.4.1 add\_resolution()

```
void PhotoFinish::Tags::add_resolution (
    Image::ptr img )
```

Definition at line 219 of file Tags.cc.

#### 6.43.4.2 add\_searchpath()

```
void PhotoFinish::Tags::add_searchpath (
    fs::path path ) [inline]
```

Definition at line 65 of file Tags.hh.

#### 6.43.4.3 copy\_from()

```
void PhotoFinish::Tags::copy_from (
    Image::ptr img )
```

Copy EXIF/IPTC/XMP tags from an image.

Definition at line 171 of file Tags.cc.

#### 6.43.4.4 copy\_to()

```
void PhotoFinish::Tags::copy_to (
    Image::ptr img ) const
```

Copy EXIF/IPTC/XMP tags to an image.

Definition at line 242 of file Tags.cc.

#### 6.43.4.5 dupe()

```
Tags::ptr PhotoFinish::Tags::dupe (
    void ) const
```

Duplicate the tags.

Definition at line 48 of file Tags.cc.

#### 6.43.4.6 EXIFtags()

```
Exiv2::ExifData& PhotoFinish::Tags::EXIFtags (
    void ) [inline]
```

The Exiv2::ExifData object.

Definition at line 71 of file Tags.hh.

#### 6.43.4.7 IPTCtags()

```
Exiv2::IptcData& PhotoFinish::Tags::IPTCtags (
    void ) [inline]
```

The Exiv2::IptcData object.

Definition at line 74 of file Tags.hh.

#### 6.43.4.8 load()

```
void PhotoFinish::Tags::load (
    fs::path filepath )
```

Load tags from supplied file path.

Definition at line 68 of file Tags.cc.

#### 6.43.4.9 make\_thumbnail()

```
void PhotoFinish::Tags::make_thumbnail (
    Image::ptr img,
    const D_thumbnail & dt )
```

Create a thumbnail from the supplied image.

Definition at line 182 of file Tags.cc.

#### 6.43.4.10 try\_load()

```
bool PhotoFinish::Tags::try_load (
    fs::path filepath )
```

Try to load tags from a file, looking in the search paths

##### Returns

if the file was found and loaded

Definition at line 57 of file Tags.cc.



#### 6.43.4.11 variables()

```
multihash& PhotoFinish::Tags::variables (
    void ) [inline]
```

The map of variables.

Definition at line 68 of file Tags.hh.

#### 6.43.4.12 XMPtags()

```
Exiv2::XmpData& PhotoFinish::Tags::XMPtags (
    void ) [inline]
```

The Exiv2::XmpData object.

Definition at line 77 of file Tags.hh.

The documentation for this class was generated from the following files:

- [Tags.hh](#)
- [Tags.cc](#)

## 6.44 PhotoFinish::Timer Class Reference

Class for doing nanosecond-accurate timings.

```
#include <Benchmark.hh>
```

### Public Member Functions

- [Timer](#) ()  
*Empty constructor.*
- void [start](#) (void)  
*Record the start time.*
- void [stop](#) (void)  
*Record the end time.*
- double [elapsed](#) (void) const  
*Return the number of seconds elapsed.*
- long long [elapsed\\_ns](#) (void) const  
*Return the number of nanoseconds elapsed.*

### 6.44.1 Detailed Description

Class for doing nanosecond-accurate timings.

Definition at line 31 of file Benchmark.hh.

## 6.44.2 Constructor & Destructor Documentation

### 6.44.2.1 Timer()

```
PhotoFinish::Timer::Timer ( )
```

Empty constructor.

Definition at line 25 of file Benchmark.cc.

## 6.44.3 Member Function Documentation

### 6.44.3.1 elapsed()

```
double PhotoFinish::Timer::elapsed (
    void ) const
```

Return the number of seconds elapsed.

Definition at line 29 of file Benchmark.cc.

### 6.44.3.2 elapsed\_ns()

```
long long PhotoFinish::Timer::elapsed_ns (
    void ) const
```

Return the number of nanoseconds elapsed.

Definition at line 36 of file Benchmark.cc.

### 6.44.3.3 start()

```
void PhotoFinish::Timer::start (
    void ) [inline]
```

Record the start time.

Definition at line 41 of file Benchmark.hh.

## 6.44.3.4 stop()

```
void PhotoFinish::Timer::stop (
    void ) [inline]
```

Record the end time.

Definition at line 47 of file Benchmark.hh.

The documentation for this class was generated from the following files:

- [Benchmark.hh](#)
- [Benchmark.cc](#)

## 6.45 CMS::Transform Class Reference

Wrap LCMS2's transform object.

```
#include <CMS.hh>
```

### Public Types

- typedef std::shared\_ptr< [Transform](#) > [ptr](#)

### Public Member Functions

- [Transform](#) ([Profile::ptr](#) input, const [Format](#) &informat, [Profile::ptr](#) output, const [Format](#) &outformat, [Intent](#) intent, cmsUInt32Number flags)  
*Construct a transform from two profiles and formats.*
- [Transform](#) (std::vector< [Profile::ptr](#) > profile, const [Format](#) &informat, const [Format](#) &outformat, [Intent](#) intent, cmsUInt32Number flags)  
*Construct a transform from multiple profiles.*
- [~Transform](#) ()  
*Destructor.*
- [Format](#) input\_format (void) const  
*Get the input format.*
- [Format](#) output\_format (void) const  
*Get the output format.*
- void [change\\_formats](#) (const [Format](#) &informat, const [Format](#) &outformat)  
*Change the input and output formats.*
- [Profile::ptr](#) device\_link (double version, cmsUInt32Number flags) const  
*Create a device link profile from this transform.*
- void [transform\\_buffer](#) (const unsigned char \*input, unsigned char \*output, cmsUInt32Number size) const

### Static Public Member Functions

- static [ptr](#) [Proofing](#) ([Profile::ptr](#) input, const [Format](#) &informat, [Profile::ptr](#) output, const [Format](#) &outformat, [Profile::ptr](#) proofing, [Intent](#) intent, [Intent](#) proofing\_intent, cmsUInt32Number flags)  
*Named constructor for creating a proofing transform.*

## Friends

- class [\\_\\_gnu\\_cxx::new\\_allocator< Transform >](#)

### 6.45.1 Detailed Description

Wrap LCMS2's transform object.

Definition at line 397 of file CMS.hh.

### 6.45.2 Member Typedef Documentation

#### 6.45.2.1 ptr

```
typedef std::shared_ptr<Transform> CMS::Transform::ptr
```

Definition at line 421 of file CMS.hh.

### 6.45.3 Constructor & Destructor Documentation

#### 6.45.3.1 Transform() [1/2]

```
CMS::Transform::Transform (
    Profile::ptr input,
    const Format & informat,
    Profile::ptr output,
    const Format & outformat,
    Intent intent,
    cmsUInt32Number flags )
```

Construct a transform from two profiles and formats.

Definition at line 545 of file CMS.cc.

#### 6.45.3.2 Transform() [2/2]

```
CMS::Transform::Transform (
    std::vector< Profile::ptr > profile,
    const Format & informat,
    const Format & outformat,
    Intent intent,
    cmsUInt32Number flags )
```

Construct a transform from multiple profiles.

Definition at line 554 of file CMS.cc.

### 6.45.3.3 ~Transform()

```
CMS::Transform::~~Transform ( )
```

Deconstructor.

Definition at line 561 of file CMS.cc.

## 6.45.4 Member Function Documentation

### 6.45.4.1 change\_formats()

```
void CMS::Transform::change_formats (
    const Format & informat,
    const Format & outformat )
```

Change the input and output formats.

Definition at line 584 of file CMS.cc.

### 6.45.4.2 device\_link()

```
Profile::ptr CMS::Transform::device_link (
    double version,
    cmsUInt32Number flags ) const
```

Create a device link profile from this transform.

Definition at line 588 of file CMS.cc.

### 6.45.4.3 input\_format()

```
Format CMS::Transform::input_format (
    void ) const
```

Get the input format.

Definition at line 576 of file CMS.cc.

#### 6.45.4.4 output\_format()

```
Format CMS::Transform::output_format (
    void ) const
```

Get the output format.

Definition at line 580 of file CMS.cc.

#### 6.45.4.5 Proofing()

```
Transform::ptr CMS::Transform::Proofing (
    Profile::ptr input,
    const Format & informat,
    Profile::ptr output,
    const Format & outformat,
    Profile::ptr proofing,
    Intent intent,
    Intent proofing_intent,
    cmsUInt32Number flags ) [static]
```

Named constructor for creating a proofing transform.

Definition at line 565 of file CMS.cc.

#### 6.45.4.6 transform\_buffer()

```
void CMS::Transform::transform_buffer (
    const unsigned char * input,
    unsigned char * output,
    cmsUInt32Number size ) const
```

Definition at line 592 of file CMS.cc.

### 6.45.5 Friends And Related Function Documentation

#### 6.45.5.1 \_\_gnu\_cxx::new\_allocator< Transform >

```
friend class __gnu_cxx::new_allocator< Transform > [friend]
```

Definition at line 405 of file CMS.hh.

The documentation for this class was generated from the following files:

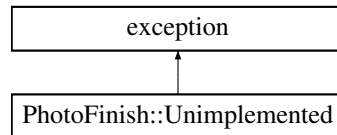
- [CMS.hh](#)
- [CMS.cc](#)

## 6.46 PhotoFinish::Unimplemented Class Reference

[Unimplemented](#) method exception.

```
#include <Exception.hh>
```

Inheritance diagram for PhotoFinish::Unimplemented:



### Public Member Functions

- [Unimplemented](#) (const std::string &c, const std::string &m)  
*Constructor.*
- virtual const char \* [what](#) () const noexcept

### Protected Attributes

- const std::string [\\_class](#)
- const std::string [\\_method](#)

#### 6.46.1 Detailed Description

[Unimplemented](#) method exception.

Definition at line 58 of file Exception.hh.

#### 6.46.2 Constructor & Destructor Documentation

##### 6.46.2.1 Unimplemented()

```
PhotoFinish::Unimplemented::Unimplemented (
    const std::string & c,
    const std::string & m ) [inline]
```

Constructor.

#### Parameters

<i>c</i>	Class name
<i>m</i>	Method name

Definition at line 68 of file Exception.hh.

### 6.46.3 Member Function Documentation

#### 6.46.3.1 what()

```
virtual const char* PhotoFinish::Unimplemented::what ( ) const [inline], [virtual], [noexcept]
```

Definition at line 72 of file Exception.hh.

### 6.46.4 Member Data Documentation

#### 6.46.4.1 \_class

```
const std::string PhotoFinish::Unimplemented::_class [protected]
```

Definition at line 60 of file Exception.hh.

#### 6.46.4.2 \_method

```
const std::string PhotoFinish::Unimplemented::_method [protected]
```

Definition at line 60 of file Exception.hh.

The documentation for this class was generated from the following file:

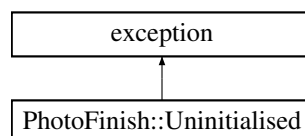
- [Exception.hh](#)

## 6.47 PhotoFinish::Uninitialised Class Reference

[Uninitialised](#) attribute exception.

```
#include <Exception.hh>
```

Inheritance diagram for PhotoFinish::Uninitialised:





## Public Member Functions

- [Uninitialised](#) (const std::string &c, const std::string &a)  
*Constructor.*
- [Uninitialised](#) (const std::string &c)  
*Constructor.*
- virtual const char \* [what](#) () const noexcept

## Protected Attributes

- const std::string [\\_class](#)
- const std::string [\\_attribute](#)

### 6.47.1 Detailed Description

[Uninitialised](#) attribute exception.

Definition at line 27 of file Exception.hh.

### 6.47.2 Constructor & Destructor Documentation

#### 6.47.2.1 Uninitialised() [1/2]

```
PhotoFinish::Uninitialised::Uninitialised (
    const std::string & c,
    const std::string & a ) [inline]
```

Constructor.

#### Parameters

<i>c</i>	Class name
<i>a</i>	Attribute name

Definition at line 37 of file Exception.hh.

#### 6.47.2.2 Uninitialised() [2/2]

```
PhotoFinish::Uninitialised::Uninitialised (
    const std::string & c ) [inline]
```

Constructor.

**Parameters**

<i>c</i>	Class name
----------	------------

Definition at line 45 of file Exception.hh.

### 6.47.3 Member Function Documentation

#### 6.47.3.1 what()

```
virtual const char* PhotoFinish::Uninitialised::what ( ) const [inline], [virtual], [noexcept]
```

Definition at line 49 of file Exception.hh.

### 6.47.4 Member Data Documentation

#### 6.47.4.1 \_attribute

```
const std::string PhotoFinish::Uninitialised::_attribute [protected]
```

Definition at line 29 of file Exception.hh.

#### 6.47.4.2 \_class

```
const std::string PhotoFinish::Uninitialised::_class [protected]
```

Definition at line 29 of file Exception.hh.

The documentation for this class was generated from the following file:

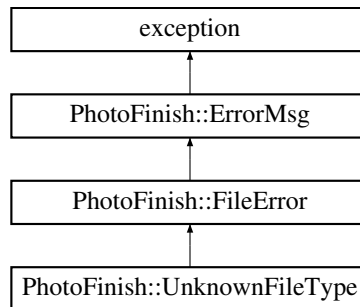
- [Exception.hh](#)

## 6.48 PhotoFinish::UnknownFileType Class Reference

Unknown file type exception.

```
#include <Exception.hh>
```

Inheritance diagram for PhotoFinish::UnknownFileType:



### Public Member Functions

- [UnknownFileType](#) (const std::string &fp, const std::string &m)  
*Constructor.*
- [UnknownFileType](#) (const std::string &fp)  
*Constructor.*
- virtual const char \* [what](#) () const noexcept

### Additional Inherited Members

#### 6.48.1 Detailed Description

Unknown file type exception.

Definition at line 176 of file Exception.hh.

#### 6.48.2 Constructor & Destructor Documentation

##### 6.48.2.1 UnknownFileType() [1/2]

```
PhotoFinish::UnknownFileType::UnknownFileType (  
    const std::string & fp,  
    const std::string & m ) [inline]
```

Constructor.

**Parameters**

<i>fp</i>	File path
<i>m</i>	Error message

Definition at line 183 of file Exception.hh.

**6.48.2.2 UnknownFileType()** [2/2]

```
PhotoFinish::UnknownFileType::UnknownFileType (
    const std::string & fp ) [inline]
```

Constructor.

**Parameters**

<i>fp</i>	File path
-----------	-----------

Definition at line 191 of file Exception.hh.

**6.48.3 Member Function Documentation****6.48.3.1 what()**

```
virtual const char* PhotoFinish::UnknownFileType::what ( ) const [inline], [virtual], [noexcept]
```

Implements [PhotoFinish::FileError](#).

Definition at line 195 of file Exception.hh.

The documentation for this class was generated from the following file:

- [Exception.hh](#)

**6.49 PhotoFinish::webp\_stream\_writer Class Reference**

A custom writer for libwebp that writes using a std::ostream object.

```
#include <WebP_ostream.hh>
```

## Public Member Functions

- [webp\\_stream\\_writer](#) (std::ostream \*s, unsigned int w, unsigned int h)  
*Constructor.*
- [~webp\\_stream\\_writer](#) ()
- void [add\\_icc](#) (CMS::Profile::ptr profile)  
*Add an LCMS2 profile to be written.*
- void [add\\_exif](#) (const Exiv2::ExifData &exif)  
*Add a set of EXIF tags to be written.*
- void [add\\_xmp](#) (const Exiv2::XmpData &xmp)  
*Add a set of XMP tags to be written.*
- void [write\\_chunk](#) (const char \*fourcc, const unsigned char \*data, unsigned int length)  
*Write a RIFF chunk.*
- void [before\\_chunk](#) (void)  
*Write stuff before a chunk is written.*
- void [modify\\_chunk](#) (unsigned char \*data)  
*Modify the current chunk.*
- void [after\\_chunk](#) (void)  
*Write stuff after a chunk has been written.*
- void [modify\\_vp8x](#) (unsigned char \*data)
- int [write](#) (unsigned char \*data, size\_t data\_size)  
*Write a block of data from the encoder.*

### 6.49.1 Detailed Description

A custom writer for libwebp that writes using a std::ostream object.

This class is so large because libwebp does not handle metadata *at all*. So we have to keep track of RIFF chunks as the encoder emits them and insert our own, even modifying one of the chunks (VP8X).

Definition at line 34 of file WebP\_ostream.hh.

### 6.49.2 Constructor & Destructor Documentation

#### 6.49.2.1 webp\_stream\_writer()

```
PhotoFinish::webp_stream_writer::webp_stream_writer (
    std::ostream * s,
    unsigned int w,
    unsigned int h )
```

Constructor.

Parameters

<i>s</i>	Pointer to a std::ostream derivative.
<i>w,h</i>	Width and height of the image

Definition at line 24 of file WebP\_ostream.cc.

#### 6.49.2.2 ~webp\_stream\_writer()

```
PhotoFinish::webp_stream_writer::~~webp_stream_writer ( )
```

Definition at line 32 of file WebP\_ostream.cc.

### 6.49.3 Member Function Documentation

#### 6.49.3.1 add\_exif()

```
void PhotoFinish::webp_stream_writer::add_exif (
    const Exiv2::ExifData & exif )
```

Add a set of EXIF tags to be written.

Definition at line 48 of file WebP\_ostream.cc.

#### 6.49.3.2 add\_icc()

```
void PhotoFinish::webp_stream_writer::add_icc (
    CMS::Profile::ptr profile )
```

Add an LCMS2 profile to be written.

Definition at line 42 of file WebP\_ostream.cc.

#### 6.49.3.3 add\_xmp()

```
void PhotoFinish::webp_stream_writer::add_xmp (
    const Exiv2::XmpData & xmp )
```

Add a set of XMP tags to be written.

Definition at line 60 of file WebP\_ostream.cc.

#### 6.49.3.4 after\_chunk()

```
void PhotoFinish::webp_stream_writer::after_chunk (
    void )
```

Write stuff after a chunk has been written.

Definition at line 107 of file WebP\_ostream.cc.

#### 6.49.3.5 before\_chunk()

```
void PhotoFinish::webp_stream_writer::before_chunk (
    void )
```

Write stuff before a chunk is written.

Definition at line 81 of file WebP\_ostream.cc.

#### 6.49.3.6 modify\_chunk()

```
void PhotoFinish::webp_stream_writer::modify_chunk (
    unsigned char * data )
```

Modify the current chunk.

Definition at line 99 of file WebP\_ostream.cc.

#### 6.49.3.7 modify\_vp8x()

```
void PhotoFinish::webp_stream_writer::modify_vp8x (
    unsigned char * data )
```

Definition at line 124 of file WebP\_ostream.cc.

#### 6.49.3.8 write()

```
int PhotoFinish::webp_stream_writer::write (
    unsigned char * data,
    size_t data_size )
```

Write a block of data from the encoder.

Definition at line 136 of file WebP\_ostream.cc.

#### 6.49.3.9 write\_chunk()

```
void PhotoFinish::webp_stream_writer::write_chunk (
    const char * fourcc,
    const unsigned char * data,
    unsigned int length )
```

Write a RIFF chunk.

Definition at line 71 of file WebP\_ostream.cc.

The documentation for this class was generated from the following files:

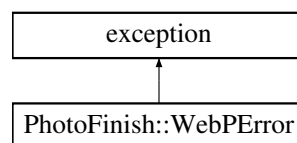
- [WebP\\_ostream.hh](#)
- [WebP\\_ostream.cc](#)

## 6.50 PhotoFinish::WebPError Class Reference

WebP exception.

```
#include <Exception.hh>
```

Inheritance diagram for PhotoFinish::WebPError:



### Public Member Functions

- [WebPError](#) (int c)  
*Constructor.*
- virtual const char \* [what](#) () const noexcept

#### 6.50.1 Detailed Description

WebP exception.

Definition at line 322 of file Exception.hh.

#### 6.50.2 Constructor & Destructor Documentation

##### 6.50.2.1 WebPError()

```
PhotoFinish::WebPError::WebPError (
    int c ) [inline]
```

Constructor.



## Parameters

<i>c</i>	Error code
----------	------------

Definition at line 331 of file Exception.hh.

### 6.50.3 Member Function Documentation

#### 6.50.3.1 what()

```
virtual const char* PhotoFinish::WebPError::what ( ) const [inline], [virtual], [noexcept]
```

Definition at line 335 of file Exception.hh.

The documentation for this class was generated from the following file:

- [Exception.hh](#)



## Chapter 7

# File Documentation

### 7.1 Benchmark.cc File Reference

```
#include "Benchmark.hh"
```

#### Namespaces

- [PhotoFinish](#)

#### Functions

- `std::ostream & PhotoFinish::operator<< (std::ostream &out, Timer t)`

### 7.2 Benchmark.hh File Reference

```
#include <ostream>
#include <time.h>
```

#### Classes

- class [PhotoFinish::Timer](#)  
*Class for doing nanosecond-accurate timings.*

#### Namespaces

- [PhotoFinish](#)

#### Functions

- `std::ostream & PhotoFinish::operator<< (std::ostream &out, Timer t)`

## Variables

- bool [PhotoFinish::benchmark\\_mode](#) = false

## 7.3 CMS.cc File Reference

```
#include <fstream>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include <string.h>
#include "CMS.hh"
```

## Namespaces

- [CMS](#)

## Macros

- #define [FLOAT\\_MASK](#) (0xffffffff ^ FLOAT\_SH(1))
- #define [OPTIMIZED\\_MASK](#) (0xffffffff ^ OPTIMIZED\_SH(1))
- #define [COLORSPACE\\_MASK](#) (0xffffffff ^ COLORSPACE\_SH(31))
- #define [SWAPFIRST\\_MASK](#) (0xffffffff ^ SWAPFIRST\_SH(1))
- #define [FLAVOR\\_MASK](#) (0xffffffff ^ FLAVOR\_SH(1))
- #define [PLANAR\\_MASK](#) (0xffffffff ^ PLANAR\_SH(1))
- #define [ENDIAN16\\_MASK](#) (0xffffffff ^ ENDIAN16\_SH(1))
- #define [DOSWAP\\_MASK](#) (0xffffffff ^ DOSWAP\_SH(1))
- #define [EXTRA\\_MASK](#) (0xffffffff ^ EXTRA\_SH(7))
- #define [CHANNELS\\_MASK](#) (0xffffffff ^ CHANNELS\_SH(15))
- #define [BYTES\\_MASK](#) (0xffffffff ^ BYTES\_SH(7))

## Functions

- std::ostream & [CMS::operator<<](#) (std::ostream &out, ColourModel model)
- std::ostream & [CMS::operator<<](#) (std::ostream &out, Format f)
- cmsIOHANDLER \* [CMS::OpenIOhandlerFromIStream](#) (std::istream \*is)
- cmsIOHANDLER \* [CMS::OpenIOhandlerFromIFStream](#) (fs::path filepath)
- cmsUInt32Number [CMS::istream\\_read](#) (cmsIOHANDLER \*iohandler, void \*Buffer, cmsUInt32Number size, cmsUInt32Number count)
- cmsBool [CMS::istream\\_seek](#) (cmsIOHANDLER \*iohandler, cmsUInt32Number offset)
- cmsBool [CMS::istream\\_close](#) (cmsIOHANDLER \*iohandler)
- cmsUInt32Number [CMS::istream\\_tell](#) (cmsIOHANDLER \*iohandler)
- cmsBool [CMS::istream\\_write](#) (cmsIOHANDLER \*iohandler, cmsUInt32Number size, const void \*Buffer)
- cmsUInt32Number [CMS::ostream\\_read](#) (cmsIOHANDLER \*iohandler, void \*Buffer, cmsUInt32Number size, cmsUInt32Number count)
- cmsBool [CMS::ostream\\_seek](#) (cmsIOHANDLER \*iohandler, cmsUInt32Number offset)
- cmsBool [CMS::ostream\\_close](#) (cmsIOHANDLER \*iohandler)
- cmsUInt32Number [CMS::ostream\\_tell](#) (cmsIOHANDLER \*iohandler)
- cmsBool [CMS::ostream\\_write](#) (cmsIOHANDLER \*iohandler, cmsUInt32Number size, const void \*Buffer)
- void [lcms2\\_errorhandler](#) (cmsContext ContextID, cmsUInt32Number ErrorCode, const char \*Text)  
*Throw a LibraryError exception when LCMS2 returns an error.*
- void [lcms2\\_error\\_adaptor](#) (void)  
*Set up an error handler with LCMS2 that will throw a LibraryError exception.*

### 7.3.1 Macro Definition Documentation

#### 7.3.1.1 BYTES\_MASK

```
#define BYTES_MASK (0xffffffff ^ BYTES_SH(7))
```

Definition at line 319 of file CMS.cc.

#### 7.3.1.2 CHANNELS\_MASK

```
#define CHANNELS_MASK (0xffffffff ^ CHANNELS_SH(15))
```

Definition at line 318 of file CMS.cc.

#### 7.3.1.3 COLORSPACE\_MASK

```
#define COLORSPACE_MASK (0xffffffff ^ COLORSPACE_SH(31))
```

Definition at line 311 of file CMS.cc.

#### 7.3.1.4 DOSWAP\_MASK

```
#define DOSWAP_MASK (0xffffffff ^ DOSWAP_SH(1))
```

Definition at line 316 of file CMS.cc.

#### 7.3.1.5 ENDIAN16\_MASK

```
#define ENDIAN16_MASK (0xffffffff ^ ENDIAN16_SH(1))
```

Definition at line 315 of file CMS.cc.

#### 7.3.1.6 EXTRA\_MASK

```
#define EXTRA_MASK (0xffffffff ^ EXTRA_SH(7))
```

Definition at line 317 of file CMS.cc.

#### 7.3.1.7 FLAVOR\_MASK

```
#define FLAVOR_MASK (0xffffffff ^ FLAVOR_SH(1))
```

Definition at line 313 of file CMS.cc.

#### 7.3.1.8 FLOAT\_MASK

```
#define FLOAT_MASK (0xffffffff ^ FLOAT_SH(1))
```

Definition at line 309 of file CMS.cc.

#### 7.3.1.9 OPTIMIZED\_MASK

```
#define OPTIMIZED_MASK (0xffffffff ^ OPTIMIZED_SH(1))
```

Definition at line 310 of file CMS.cc.

#### 7.3.1.10 PLANAR\_MASK

```
#define PLANAR_MASK (0xffffffff ^ PLANAR_SH(1))
```

Definition at line 314 of file CMS.cc.

#### 7.3.1.11 SWAPFIRST\_MASK

```
#define SWAPFIRST_MASK (0xffffffff ^ SWAPFIRST_SH(1))
```

Definition at line 312 of file CMS.cc.

### 7.3.2 Function Documentation

#### 7.3.2.1 lcms2\_error\_adaptor()

```
void lcms2_error_adaptor (
    void )
```

Set up an error handler with LCMS2 that will throw a `LibraryError` exception.

Definition at line 710 of file `CMS.cc`.

#### 7.3.2.2 lcms2\_errorhandler()

```
void lcms2_errorhandler (
    cmsContext ContextID,
    cmsUInt32Number ErrorCode,
    const char * Text )
```

Throw a `LibraryError` exception when LCMS2 returns an error.

Definition at line 706 of file `CMS.cc`.

## 7.4 CMS.hh File Reference

```
#include <istream>
#include <ostream>
#include <memory>
#include <boost/filesystem.hpp>
#include <lcms2.h>
#include <lcms2_plugin.h>
#include "Exception.hh"
```

### Classes

- class [CMS::Profile](#)  
*Wrap LCMS2's cmsHPROFILE.*
- class [CMS::Format](#)  
*Wrap LCMS2's pixel format.*
- class [CMS::Transform](#)  
*Wrap LCMS2's transform object.*

### Namespaces

- [CMS](#)

## Enumerations

- enum `CMS::ColourModel` {  
`CMS::ColourModel::Any` = 0, `CMS::ColourModel::Greyscale` = 3, `CMS::ColourModel::RGB`, `CMS::ColourModel::CMY`,  
`CMS::ColourModel::CMYK`, `CMS::ColourModel::YCbCr`, `CMS::ColourModel::YUV`, `CMS::ColourModel::XYZ`,  
`CMS::ColourModel::Lab`, `CMS::ColourModel::YUVK`, `CMS::ColourModel::HSV`, `CMS::ColourModel::HLS`,  
`CMS::ColourModel::Yxy`, `CMS::ColourModel::MCH1`, `CMS::ColourModel::MCH2`, `CMS::ColourModel::MCH3`,  
`CMS::ColourModel::MCH4`, `CMS::ColourModel::MCH5`, `CMS::ColourModel::MCH6`, `CMS::ColourModel::MCH7`,  
`CMS::ColourModel::MCH8`, `CMS::ColourModel::MCH9`, `CMS::ColourModel::MCH10`, `CMS::ColourModel::MCH11`,  
`CMS::ColourModel::MCH12`, `CMS::ColourModel::MCH13`, `CMS::ColourModel::MCH14`, `CMS::ColourModel::MCH15`,  
`CMS::ColourModel::LabV2` }

*An enum class of LCMS2's colour models.*

- enum `CMS::Intent` {  
`CMS::Intent::Perceptual`, `CMS::Intent::Relative_colormetric`, `CMS::Intent::Saturation`, `CMS::Intent::Absolute_colormetric`,  
`CMS::Intent::Preserve_k_only_perceptual` = 10, `CMS::Intent::Preserve_k_only_relative_colormetric`, `CMS::Intent::Preserve_k_only_saturation`, `CMS::Intent::Preserve_k_only_absolute_colormetric`,  
`CMS::Intent::Preserve_k_plane_perceptual`, `CMS::Intent::Preserve_k_plane_relative_colormetric`, `CMS::Intent::Preserve_k_plane_saturation`, `CMS::Intent::Preserve_k_plane_absolute_colormetric` }

*Wrap LCMS2's intents.*

## Functions

- `std::ostream & CMS::operator<< (std::ostream &out, ColourModel model)`
- `std::ostream & CMS::operator<< (std::ostream &out, Format f)`
- `cmsIOHANDLER * CMS::OpenIOhandlerFromIStream (std::istream *is)`
- `cmsIOHANDLER * CMS::OpenIOhandlerFromIFStream (fs::path filepath)`
- `cmsUInt32Number CMS::istream_read (cmsIOHANDLER *iohandler, void *Buffer, cmsUInt32Number size, cmsUInt32Number count)`
- `cmsBool CMS::istream_seek (cmsIOHANDLER *iohandler, cmsUInt32Number offset)`
- `cmsBool CMS::istream_close (cmsIOHANDLER *iohandler)`
- `cmsUInt32Number CMS::istream_tell (cmsIOHANDLER *iohandler)`
- `cmsBool CMS::istream_write (cmsIOHANDLER *iohandler, cmsUInt32Number size, const void *Buffer)`
- `cmsUInt32Number CMS::ostream_read (cmsIOHANDLER *iohandler, void *Buffer, cmsUInt32Number size, cmsUInt32Number count)`
- `cmsBool CMS::ostream_seek (cmsIOHANDLER *iohandler, cmsUInt32Number offset)`
- `cmsBool CMS::ostream_close (cmsIOHANDLER *iohandler)`
- `cmsUInt32Number CMS::ostream_tell (cmsIOHANDLER *iohandler)`
- `cmsBool CMS::ostream_write (cmsIOHANDLER *iohandler, cmsUInt32Number size, const void *Buffer)`
- `void lcms2_error_adaptor (void)`

*Set up an error handler with LCMS2 that will throw a `LibraryError` exception.*

### 7.4.1 Function Documentation



#### 7.4.1.1 lcms2\_error\_adaptor()

```
void lcms2_error_adaptor (
    void )
```

Set up an error handler with LCMS2 that will throw a LibraryError exception.

Definition at line 710 of file CMS.cc.

## 7.5 CropSolution.cc File Reference

```
#include <boost/lexical_cast.hpp>
#include <math.h>
#include <omp.h>
#include "CropSolution.hh"
```

### Namespaces

- [PhotoFinish](#)

### Macros

- #define [sqr](#)(x) ((x) \* (x))
- #define [min](#)(a, b) ((a) < (b) ? (a) : (b))
- #define [max](#)(a, b) ((a) > (b) ? (a) : (b))

### Functions

- void [PhotoFinish::add\\_rulers](#) (multihash &vars, std::string key, rulerlist &rulers)  
*Parse named variables into a list of rulers.*
- void [PhotoFinish::add\\_ruler\\_pins](#) (rulerlist &rulers, unsigned int [max](#))  
*Add rulers to the either side of an image if there aren't enough.*

### 7.5.1 Macro Definition Documentation

#### 7.5.1.1 max

```
#define max(  
    a,  
    b ) ((a) > (b) ? (a) : (b))
```

Definition at line 26 of file CropSolution.cc.

### 7.5.1.2 min

```
#define min(  
    a,  
    b ) ((a) < (b) ? (a) : (b))
```

Definition at line 25 of file CropSolution.cc.

### 7.5.1.3 sqr

```
#define sqr(  
    x ) ((x) * (x))
```

Definition at line 24 of file CropSolution.cc.

## 7.6 CropSolution.hh File Reference

```
#include <map>  
#include <memory>  
#include <ostream>  
#include <string>  
#include <utility>  
#include <list>  
#include "Frame.hh"
```

### Classes

- class [PhotoFinish::CropSolver](#)  
*Class for finding the best frame position for cropping.*

### Namespaces

- [PhotoFinish](#)

### Typedefs

- typedef std::pair< double, double > [PhotoFinish::rulerpair](#)  
*Ruler paramaters - percentage of final image vs. pixel position in original.*
- typedef std::list< rulerpair > [PhotoFinish::rulerlist](#)  
*A list of rulers.*

## 7.7 Definable.hh File Reference

```
#include <ostream>
#include <string>
```

### Classes

- class [PhotoFinish::definable< T >](#)  
*Template class for storing things that can be defined or undefined.*
- class [PhotoFinish::Role\\_Definable](#)  
*Base class for adding "definable" attribute.*

### Namespaces

- [PhotoFinish](#)

## 7.8 Destination.cc File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <memory>
#include <boost/lexical_cast.hpp>
#include <boost/algorithm/string.hpp>
#include <string.h>
#include "Destination_items.hh"
#include "Destination.hh"
#include "CropSolution.hh"
#include "ImageFile.hh"
#include "Exception.hh"
```

### Namespaces

- [PhotoFinish](#)

## 7.9 Destination.hh File Reference

```
#include "yaml-cpp/yaml.h"
#include <string>
#include <map>
#include <boost/filesystem.hpp>
#include "CMS.hh"
#include "Destination_items.hh"
#include "Image.hh"
#include "Frame.hh"
#include "Definable.hh"
```

## Classes

- class [PhotoFinish::Destination](#)  
*Represents a destination, read from destinations.yml.*
- class [PhotoFinish::Destinations](#)  
*A wrapper class for reading destinations from a YAML file and storing them in a map.*

## Namespaces

- [PhotoFinish](#)

## 7.10 Destination\_items.cc File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <boost/lexical_cast.hpp>
#include <boost/algorithm/string.hpp>
#include <string.h>
#include "Destination_items.hh"
#include "Destination.hh"
#include "CropSolution.hh"
#include "Exception.hh"
```

## Namespaces

- [PhotoFinish](#)

## 7.11 Destination\_items.hh File Reference

```
#include <string>
#include <memory>
#include <vector>
#include "yaml-cpp/yaml.h"
#include <boost/filesystem.hpp>
#include "CMS.hh"
#include "Image.hh"
#include "Definable.hh"
```

## Classes

- class [PhotoFinish::D\\_sharpen](#)  
*Sharpen parameters for destination.*
- class [PhotoFinish::D\\_resize](#)  
*Resize parameters for destination.*
- class [PhotoFinish::D\\_target](#)  
*Target parameters for destination.*
- class [PhotoFinish::D\\_JPEG](#)  
*JPEG parameters for destination.*
- class [PhotoFinish::D\\_PNG](#)  
*PNG parameters for destination.*
- class [PhotoFinish::D\\_TIFF](#)  
*TIFF parameters for destination.*
- class [PhotoFinish::D\\_JP2](#)  
*JP2 parameters for destination.*
- class [PhotoFinish::D\\_WebP](#)  
*WebP parameters for destination.*
- class [PhotoFinish::D\\_JXR](#)  
*JPEG XR parameters for destination.*
- class [PhotoFinish::D\\_profile](#)  
*ICC profile parameters for destination.*
- class [PhotoFinish::D\\_thumbnail](#)  
*Thumbnail parameters for destination.*

## Namespaces

- [PhotoFinish](#)

## Typedefs

- typedef std::map< std::string, std::string > [PhotoFinish::hash](#)  
*A simple hash.*
- typedef std::vector< std::string > [PhotoFinish::stringlist](#)  
*A list of strings.*
- typedef std::map< std::string, stringlist > [PhotoFinish::multihash](#)  
*A hash of string lists.*

## 7.12 Ditherer.cc File Reference

```
#include <vector>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "Ditherer.hh"
#include "sample.h"
```

## Namespaces

- [PhotoFinish](#)

## Macros

- `#define pos ((x * _channels) + c)`
- `#define prevpos (((x - 1) * _channels) + c)`
- `#define nextpos (((x + 1) * _channels) + c)`

### 7.12.1 Macro Definition Documentation

#### 7.12.1.1 nextpos

```
#define nextpos (((x + 1) * _channels) + c)
```

Definition at line 85 of file Ditherer.cc.

#### 7.12.1.2 pos

```
#define pos ((x * _channels) + c)
```

Definition at line 83 of file Ditherer.cc.

#### 7.12.1.3 prevpos

```
#define prevpos (((x - 1) * _channels) + c)
```

Definition at line 84 of file Ditherer.cc.

## 7.13 Ditherer.hh File Reference

```
#include <lcms2.h>  
#include "sample.h"
```

## Classes

- class [PhotoFinish::Ditherer](#)  
*Class for dithering images down to 8-bit components.*

## Namespaces

- [PhotoFinish](#)

## 7.14 Exception.hh File Reference

```
#include <string>
#include <exception>
```

## Classes

- class [PhotoFinish::Uninitialised](#)  
*Uninitialised attribute exception.*
- class [PhotoFinish::Unimplemented](#)  
*Unimplemented method exception.*
- class [PhotoFinish::NoResults](#)  
*No results exception.*
- class [PhotoFinish::NoTargets](#)  
*No targets exception.*
- class [PhotoFinish::ErrorMsg](#)  
*Generic error message exception.*
- class [PhotoFinish::MemAllocError](#)  
*Memory allocation exception.*
- class [PhotoFinish::FileError](#)  
*File error abstract base exception.*
- class [PhotoFinish::UnknownFileType](#)  
*Unknown file type exception.*
- class [PhotoFinish::FileOpenError](#)  
*File open exception.*
- class [PhotoFinish::FileContentError](#)  
*File content exception.*
- class [PhotoFinish::DestinationError](#)  
*Destination exception.*
- class [PhotoFinish::LibraryError](#)  
*Library exception.*
- class [PhotoFinish::cmsTypeError](#)
- class [PhotoFinish::WebPError](#)  
*WebP exception.*

## Namespaces

- [PhotoFinish](#)

## 7.15 Frame.cc File Reference

```
#include <iostream>
#include <iomanip>
#include <omp.h>
#include "Frame.hh"
#include "Destination_items.hh"
#include "Kernel1Dvar.hh"
```

### Namespaces

- [PhotoFinish](#)

## 7.16 Frame.hh File Reference

```
#include <memory>
#include "Destination_items.hh"
```

### Classes

- class [PhotoFinish::Frame](#)  
*Crop+rescaling parameters.*

### Namespaces

- [PhotoFinish](#)

## 7.17 Image.cc File Reference

```
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <omp.h>
#include "Image.hh"
#include "ImageFile.hh"
#include "Benchmark.hh"
```

### Namespaces

- [PhotoFinish](#)



## Functions

- `template<typename A , typename B >`  
`void PhotoFinish::transfer_alpha_typed2` (unsigned int width, unsigned char src\_channels, const A \*src\_row, unsigned char dest\_channels, const B \*dest\_row)
- `template<typename A >`  
`void PhotoFinish::transfer_alpha_typed` (unsigned int width, unsigned char src\_channels, const A \*src\_row, CMS::Format dest\_format, const unsigned char \*dest\_row)
- `void PhotoFinish::transfer_alpha` (unsigned int width, CMS::Format src\_format, const unsigned char \*src\_row, CMS::Format dest\_format, const unsigned char \*dest\_row)
- `std::string PhotoFinish::profile_name` (CMS::Profile::ptr profile)

## 7.18 Image.hh File Reference

```
#include <memory>
#include <exiv2/exiv2.hpp>
#include "Definable.hh"
#include "CMS.hh"
#include "sample.h"
```

## Classes

- class `PhotoFinish::Image`  
*An image class.*

## Namespaces

- `PhotoFinish`

## Functions

- `template<typename T >`  
`T PhotoFinish::scaleval` (void)  
*A template function that returns the 'scale' value of a type.*
- `template<>`  
`unsigned char PhotoFinish::scaleval< unsigned char >` (void)
- `template<>`  
`unsigned short int PhotoFinish::scaleval< unsigned short int >` (void)
- `template<>`  
`unsigned int PhotoFinish::scaleval< unsigned int >` (void)
- `template<>`  
`unsigned long long PhotoFinish::scaleval< unsigned long long >` (void)
- `template<>`  
`float PhotoFinish::scaleval< float >` (void)
- `template<>`  
`double PhotoFinish::scaleval< double >` (void)
- `template<typename T >`  
`T PhotoFinish::limitval` (SAMPLE v)  
*A template function that limits a floating-point value while converting to another type.*

- `template<>`  
    `unsigned char PhotoFinish::limitval< unsigned char > (SAMPLE v)`
- `template<>`  
    `unsigned short int PhotoFinish::limitval< unsigned short int > (SAMPLE v)`
- `template<>`  
    `unsigned int PhotoFinish::limitval< unsigned int > (SAMPLE v)`
- `template<>`  
    `unsigned long long PhotoFinish::limitval< unsigned long long > (SAMPLE v)`
- `template<>`  
    `float PhotoFinish::limitval< float > (SAMPLE v)`
- `template<>`  
    `double PhotoFinish::limitval< double > (SAMPLE v)`

## 7.19 ImageFile.cc File Reference

```
#include <boost/algorithm/string.hpp>
#include <boost/filesystem.hpp>
#include "ImageFile.hh"
#include "Exception.hh"
```

### Namespaces

- [PhotoFinish](#)

## 7.20 ImageFile.hh File Reference

```
#include <string>
#include <memory>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include "CMS.hh"
#include "Image.hh"
#include "Destination.hh"
#include "Exception.hh"
#include "sample.h"
```

### Classes

- class [PhotoFinish::ImageFilepath](#)  
    *Class for holding filename and the image format.*
- class [PhotoFinish::ImageReader](#)  
    *Abstract base class for reading image files.*
- class [PhotoFinish::ImageWriter](#)  
    *Abstract base class for writing image files.*
- class [PhotoFinish::SOLwriter](#)  
    *Write the boot logo files for use on Motorola Atrix 4G and possibly other phones.*

## Namespaces

- [PhotoFinish](#)

## Functions

- bool [PhotoFinish::exists](#) (const ImageFilepath &fp)
- std::time\_t [PhotoFinish::last\\_write\\_time](#) (const ImageFilepath &fp)

## 7.21 JP2.hh File Reference

```
#include <openjpeg.h>
```

## Namespaces

- [PhotoFinish](#)

## Functions

- void [PhotoFinish::error\\_callback](#) (const char \*msg, void \*client\_data)  
*Error callback for OpenJPEG - throw a [LibraryError](#) exception.*
- void [PhotoFinish::warning\\_callback](#) (const char \*msg, void \*client\_data)  
*Warning callback for OpenJPEG - print the message to STDERR.*
- void [PhotoFinish::info\\_callback](#) (const char \*msg, void \*client\_data)  
*Info callback for OpenJPEG - print the indented message to STDERR.*
- template<typename T >  
void [PhotoFinish::read\\_planar](#) (unsigned int width, unsigned char channels, opj\_image\_t \*image, T \*row, unsigned int y)  
*Read a row of image data from OpenJPEG's planar integer components into an LCMS2-compatible single array.*
- template<typename T >  
void [PhotoFinish::write\\_planar](#) (unsigned int width, unsigned char channels, T \*row, opj\_image\_t \*image, unsigned int y)  
*Read a row of planar pixel data into OpenJPEG's planar components.*
- template<typename T >  
void [PhotoFinish::write\\_packed](#) (unsigned int width, unsigned char channels, T \*row, opj\_image\_t \*image, unsigned int y)  
*Read a row of packed pixel data into OpenJPEG's planar components.*

## 7.22 JP2\_callbacks.cc File Reference

```
#include <iostream>
#include <string.h>
#include "Exception.hh"
```

## Namespaces

- [PhotoFinish](#)

## Functions

- void [PhotoFinish::error\\_callback](#) (const char \*msg, void \*client\_data)  
*Error callback for OpenJPEG - throw a [LibraryError](#) exception.*
- void [PhotoFinish::warning\\_callback](#) (const char \*msg, void \*client\_data)  
*Warning callback for OpenJPEG - print the message to STDERR.*
- void [PhotoFinish::info\\_callback](#) (const char \*msg, void \*client\_data)  
*Info callback for OpenJPEG - print the indented message to STDERR.*

## 7.23 JP2reader.cc File Reference

```
#include <boost/algorithm/string.hpp>
#include <openjpeg.h>
#include <omp.h>
#include "ImageFile.hh"
#include "Exception.hh"
#include "JP2.hh"
```

## Namespaces

- [PhotoFinish](#)

## 7.24 JP2writer.cc File Reference

```
#include <boost/algorithm/string.hpp>
#include <openjpeg.h>
#include <omp.h>
#include "ImageFile.hh"
#include "Exception.hh"
#include "JP2.hh"
```

## Namespaces

- [PhotoFinish](#)

## 7.25 JPEG.hh File Reference

```
#include <stdio.h>
#include <jpeglib.h>
#include "CMS.hh"
#include "Image.hh"
#include "Destination.hh"
```

## Namespaces

- [PhotoFinish](#)

## Functions

- void [PhotoFinish::jpeg\\_istream\\_src](#) (j\_decompress\_ptr dinfo, std::istream \*is)  
*Set up a "source manager" on the given JPEG decompression structure to read from an istream.*
- void [PhotoFinish::jpeg\\_istream\\_src\\_free](#) (j\_decompress\_ptr dinfo)  
*Free the data structures of the istream source manager.*
- void [PhotoFinish::jpeg\\_ostream\\_dest](#) (j\_compress\_ptr cinfo, std::ostream \*os)  
*Setup a "destination manager" on the given JPEG compression structure to write to an ostream.*
- void [PhotoFinish::jpeg\\_ostream\\_dest\\_free](#) (j\_compress\_ptr cinfo)  
*Free the data structures of the ostream destination manager.*
- void [PhotoFinish::jpegfile\\_scan\\_RGB](#) (jpeg\_compress\_struct \*cinfo)  
*Create a scan "script" for an RGB image.*
- void [PhotoFinish::jpegfile\\_scan\\_greyscale](#) (jpeg\_compress\_struct \*cinfo)  
*Create a scan "script" for a greyscale image.*
- CMS::Profile::ptr [PhotoFinish::jpeg\\_read\\_profile](#) (jpeg\_decompress\_struct \*dinfo, Destination::ptr dest)  
*Read an ICC profile from APP2 markers in a JPEG file.*
- void [PhotoFinish::jpeg\\_write\\_profile](#) (jpeg\_compress\_struct \*cinfo, unsigned char \*data, unsigned int size)  
*Write an ICC profile into APP2 markers in a JPEG file.*

## 7.26 JPEG\_iostream.cc File Reference

```
#include <iostream>
#include <stdio.h>
#include <jpeglib.h>
#include "Exception.hh"
```

## Classes

- struct [PhotoFinish::jpeg\\_source\\_state\\_t](#)  
*Structure holding information for the istream reader.*
- struct [PhotoFinish::jpeg\\_destination\\_state\\_t](#)  
*Structure holding information for the ostream writer.*

## Namespaces

- [PhotoFinish](#)

## Functions

- void [PhotoFinish::jpeg\\_istream\\_init\\_source](#) (j\_decompress\_ptr dinfo)  
*Initialise the istream source manager.*
- boolean [PhotoFinish::jpeg\\_istream\\_fill\\_input\\_buffer](#) (j\_decompress\_ptr dinfo)  
*Fill the buffer.*
- void [PhotoFinish::jpeg\\_istream\\_skip\\_input\\_data](#) (j\_decompress\_ptr dinfo, long num\_bytes)  
*Skip some data.*
- boolean [PhotoFinish::jpeg\\_istream\\_resync\\_to\\_restart](#) (j\_decompress\_ptr dinfo, int desired)  
*Resync to start?!?*
- void [PhotoFinish::jpeg\\_istream\\_term\\_source](#) (j\_decompress\_ptr dinfo)  
*Terminate the istream source manager.*
- void [PhotoFinish::jpeg\\_istream\\_src](#) (j\_decompress\_ptr dinfo, std::istream \*is)  
*Set up a "source manager" on the given JPEG decompression structure to read from an istream.*
- void [PhotoFinish::jpeg\\_istream\\_src\\_free](#) (j\_decompress\_ptr dinfo)  
*Free the data structures of the istream source manager.*
- void [PhotoFinish::jpeg\\_ostream\\_dest](#) (j\_compress\_ptr cinfo, std::ostream \*os)  
*Setup a "destination manager" on the given JPEG compression structure to write to an ostream.*
- void [PhotoFinish::jpeg\\_ostream\\_dest\\_free](#) (j\_compress\_ptr cinfo)  
*Free the data structures of the ostream destination manager.*

## 7.27 JPEG\_profiles.cc File Reference

```
#include <iostream>
#include <map>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <jpeglib.h>
#include "CMS.hh"
#include "Destination.hh"
```

## Namespaces

- [PhotoFinish](#)

## Functions

- [CMS::Profile::ptr PhotoFinish::jpeg\\_read\\_profile](#) (jpeg\_decompress\_struct \*dinfo, Destination::ptr dest)  
*Read an ICC profile from APP2 markers in a JPEG file.*
- void [PhotoFinish::jpeg\\_write\\_profile](#) (jpeg\_compress\_struct \*cinfo, unsigned char \*data, unsigned int size)  
*Write an ICC profile into APP2 markers in a JPEG file.*

## 7.28 JPEG\_scans.cc File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <jpeglib.h>
```

### Namespaces

- [PhotoFinish](#)

### Functions

- void [PhotoFinish::jpegfile\\_scan\\_RGB](#) (jpeg\_compress\_struct \*cinfo)  
*Create a scan "script" for an RGB image.*
- void [PhotoFinish::jpegfile\\_scan\\_greyscale](#) (jpeg\_compress\_struct \*cinfo)  
*Create a scan "script" for a greyscale image.*

## 7.29 JPEGreader.cc File Reference

```
#include <iostream>
#include <queue>
#include <list>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include <string.h>
#include <stdio.h>
#include <jpeglib.h>
#include <omp.h>
#include "ImageFile.hh"
#include "Image.hh"
#include "JPEG.hh"
```

### Namespaces

- [PhotoFinish](#)

## 7.30 JPEGwriter.cc File Reference

```
#include <iostream>
#include <queue>
#include <list>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include <string.h>
#include <stdio.h>
#include <jpeglib.h>
#include <omp.h>
#include "ImageFile.hh"
#include "Image.hh"
#include "JPEG.hh"
```

## Namespaces

- [PhotoFinish](#)

## Functions

- void [PhotoFinish::jpeg\\_error\\_exit](#) (j\_common\_ptr cinfo)

## 7.31 JXR.hh File Reference

```
#include <JXRGlue.h>
#include <vector>
#include <lcms2.h>
#include "CMS.hh"
```

## Namespaces

- [PhotoFinish](#)

## Macros

- #define [JXRcheck](#)(exp) if ((rc = (exp)) < 0) throw LibraryError("jxrlib", std::string(#exp) + " returned " + std::string(rc))
- #define [FmtPair](#)(n, g) std::make\_pair<unsigned int, const PKPixelFormatGUID\*>(n, g)

## Typedefs

- typedef std::vector< std::pair< unsigned int, const PKPixelFormatGUID \* > > [PhotoFinish::jxr\\_format\\_subst](#)

## Functions

- const PKPixelFormatGUID & [PhotoFinish::jxr\\_pixel\\_format](#) (unsigned int n)
- [CMS::Format](#) [PhotoFinish::jxr\\_cms\\_format](#) (const PKPixelFormatGUID &g)

### 7.31.1 Macro Definition Documentation

#### 7.31.1.1 FmtPair

```
#define FmtPair(
    n,
    g ) std::make_pair<unsigned int, const PKPixelFormatGUID*>(n, g)
```

Definition at line 36 of file JXR.hh.



## 7.31.1.2 JXRcheck

```
#define JXRcheck(
    exp ) if ((rc = (exp)) < 0) throw LibraryError("jxrlib", std::string(#exp) + "
returned " + std::to_string(rc))
```

Definition at line 30 of file JXR.hh.

## 7.32 JXR\_formats.cc File Reference

```
#include <iostream>
#include "Exception.hh"
#include "JXR.hh"
```

## Namespaces

- [PhotoFinish](#)

## Functions

- const PKPixelFormatGUID & [PhotoFinish::jxr\\_pixel\\_format](#) (unsigned int n)
- [CMS::Format](#) [PhotoFinish::jxr\\_cms\\_format](#) (const PKPixelFormatGUID &g)

## Variables

- jxr\_format\_subst [PhotoFinish::JXR\\_format\\_table](#)

## 7.33 JXRreader.cc File Reference

```
#include <JXRGlue.h>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include "ImageFile.hh"
#include "Image.hh"
#include "JXR.hh"
```

## Namespaces

- [PhotoFinish](#)

## Macros

- #define [jxr\\_metadata\\_size](#)(decoder, name) decoder->WMP.wmiDEMisc.u##name##ByteCount
- #define [jxr\\_metadata\\_data](#)(decoder, name, data)

### 7.33.1 Macro Definition Documentation

#### 7.33.1.1 jxr\_metadata\_data

```
#define jxr_metadata_data(  
    decoder,  
    name,  
    data )
```

#### Value:

```
struct WMPStream* s = decoder->pStream;          \  
    size_t curr_pos;                             \  
    JXRcheck(s->GetPos(s, &curr_pos));           \  
    JXRcheck(s->SetPos(s, decoder->WMP.wmiDEMisc.u##name##Offset)); \  
    JXRcheck(s->Read(s, data, decoder->WMP.wmiDEMisc.u##name##ByteCount)); \  
    JXRcheck(s->SetPos(s, curr_pos));
```

Definition at line 36 of file JXRreader.cc.

#### 7.33.1.2 jxr\_metadata\_size

```
#define jxr_metadata_size(  
    decoder,  
    name ) decoder->WMP.wmiDEMisc.u##name##ByteCount
```

Definition at line 35 of file JXRreader.cc.

## 7.34 JXRwriter.cc File Reference

```
#include <JXRGlue.h>  
#include <boost/algorithm/string.hpp>  
#include <boost/filesystem.hpp>  
#include <boost/filesystem/fstream.hpp>  
#include "ImageFile.hh"  
#include "Image.hh"  
#include "JXR.hh"
```

### Namespaces

- [PhotoFinish](#)

## 7.35 Kernel1Dvar.cc File Reference

```
#include <iostream>
#include <iomanip>
#include <boost/algorithm/string.hpp>
#include <stdlib.h>
#include <math.h>
#include <omp.h>
#include "Benchmark.hh"
#include "Kernel1Dvar.hh"
```

### Namespaces

- [PhotoFinish](#)

### Macros

- `#define sqr(x) ((x) * (x))`
- `#define min(x, y) ((x) < (y) ? (x) : (y))`

#### 7.35.1 Macro Definition Documentation

##### 7.35.1.1 min

```
#define min(  
    x,  
    y ) ((x) < (y) ? (x) : (y))
```

Definition at line 29 of file Kernel1Dvar.cc.

##### 7.35.1.2 sqr

```
#define sqr(  
    x ) ((x) * (x))
```

Definition at line 28 of file Kernel1Dvar.cc.

## 7.36 Kernel1Dvar.hh File Reference

```
#include <memory>
#include "Destination_items.hh"
#include "Exception.hh"
#include "Definable.hh"
#include "sample.h"
```

## Classes

- class [PhotoFinish::Kernel1Dvar](#)  
*Creates and stores coefficients for cropping and resizing an image.*
- class [PhotoFinish::Lanczos](#)  
*Lanczos filter.*

## Namespaces

- [PhotoFinish](#)

## 7.37 Kernel2D.cc File Reference

```
#include <stdlib.h>
#include <omp.h>
#include "Kernel2D.hh"
#include "Destination_items.hh"
#include "Benchmark.hh"
```

## Namespaces

- [PhotoFinish](#)

## Macros

- `#define sqr(x) ((x) * (x))`

### 7.37.1 Macro Definition Documentation

#### 7.37.1.1 `sqr`

```
#define sqr(  
    x ) ((x) * (x))
```

Definition at line 25 of file Kernel2D.cc.

## 7.38 Kernel2D.hh File Reference

```
#include <memory>
#include "Image.hh"
#include "Exception.hh"
#include "Definable.hh"
#include "sample.h"
```

## Classes

- class [PhotoFinish::Kernel2D](#)  
*Creates and stores coefficients for convolving an image.*
- class [PhotoFinish::GaussianSharpen](#)  
*GaussianSharpen kernel.*

## Namespaces

- [PhotoFinish](#)

## 7.39 LCMS2ErrorHandler.cc File Reference

```
#include <lcms2.h>
#include "Exception.hh"
```

## Namespaces

- [PhotoFinish](#)

## Functions

- void [PhotoFinish::lcms2\\_errorhandler](#) (cmsContext ContextID, cmsUInt32Number ErrorCode, const char \*Text)  
*Throw a [LibraryError](#) exception when LCMS2 returns an error.*
- void [PhotoFinish::lcms2\\_error\\_adaptor](#) (void)  
*Set up an error handler with LCMS2 that will throw a [LibraryError](#) exception.*

## 7.40 photofinish.cc File Reference

```
#include <iostream>
#include <string>
#include <deque>
#include <boost/filesystem.hpp>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include "Image.hh"
#include "ImageFile.hh"
#include "Destination.hh"
#include "Tags.hh"
#include "Kernel2D.hh"
#include "Exception.hh"
#include "Benchmark.hh"
```

## Functions

- [int main](#) (int argc, char \*argv[])

### 7.40.1 Function Documentation

#### 7.40.1.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Definition at line 38 of file photofinish.cc.

## 7.41 PNGreader.cc File Reference

```
#include <errno.h>
#include <png.h>
#include <zlib.h>
#include <time.h>
#include <omp.h>
#include <boost/algorithm/string.hpp>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include <iostream>
#include "ImageFile.hh"
#include "Image.hh"
#include "PNGreader_cb.hh"
```

## Namespaces

- [PhotoFinish](#)

## 7.42 PNGreader\_cb.cc File Reference

```
#include "PNGreader_cb.hh"
```

## Namespaces

- [PhotoFinish](#)

## Functions

- void [PhotoFinish::png\\_info\\_cb](#) (png\_structp png, png\_infop info)  
*Called by libPNG when the iHDR chunk has been read with the main "header" information.*
- void [PhotoFinish::png\\_row\\_cb](#) (png\_structp png, png\_bytep row\_data, png\_uint\_32 row\_num, int pass)  
*Called by libPNG when a row of image data has been read.*
- void [PhotoFinish::png\\_end\\_cb](#) (png\_structp png, png\_infop info)  
*Called by libPNG when the image data has finished.*

## 7.43 PNGreader\_cb.hh File Reference

```
#include <png.h>
#include "Destination.hh"
#include "Image.hh"
```

## Classes

- struct [PhotoFinish::PNGreader\\_cb](#)

## Namespaces

- [PhotoFinish](#)

## Functions

- void [PhotoFinish::png\\_info\\_cb](#) (png\_structp png, png\_infop info)  
*Called by libPNG when the iHDR chunk has been read with the main "header" information.*
- void [PhotoFinish::png\\_row\\_cb](#) (png\_structp png, png\_bytep row\_data, png\_uint\_32 row\_num, int pass)  
*Called by libPNG when a row of image data has been read.*
- void [PhotoFinish::png\\_end\\_cb](#) (png\_structp png, png\_infop info)  
*Called by libPNG when the image data has finished.*

## 7.44 PNGwriter.cc File Reference

```
#include <errno.h>
#include <png.h>
#include <zlib.h>
#include <time.h>
#include <omp.h>
#include <boost/algorithm/string.hpp>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include <iostream>
#include "ImageFile.hh"
#include "Image.hh"
```

## Namespaces

- [PhotoFinish](#)

## Functions

- void [PhotoFinish::png\\_write\\_ostream\\_cb](#) (png\_structp png, png\_bytep buffer, png\_size\_t length)  
*libPNG callback for writing to an ostream*
- void [PhotoFinish::png\\_flush\\_ostream\\_cb](#) (png\_structp png)  
*libPNG callback for flushing an ostream*

## 7.45 process\_scans.cc File Reference

```
#include <boost/program_options/options_description.hpp>
#include <boost/program_options/variables_map.hpp>
#include <boost/program_options/parsers.hpp>
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>
#include <boost/filesystem.hpp>
#include <boost/algorithm/string.hpp>
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include "CMS.hh"
#include "Image.hh"
#include "ImageFile.hh"
#include "Destination.hh"
#include "Tags.hh"
#include "Kernel2D.hh"
#include "Exception.hh"
#include "Benchmark.hh"
```

## Functions

- void [make\\_preview](#) ([Image::ptr](#) orig\_image, [Destination::ptr](#) orig\_dest, [Tags::ptr](#) filetags, [ImageWriter::ptr](#) preview\_file, bool can\_free=false)
- void [preview\\_dir](#) (fs::path dir, std::string format, std::shared\_ptr< [Tags](#) > tags)
- int [main](#) (int argc, char \*argv[])

### 7.45.1 Function Documentation



#### 7.45.1.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Definition at line 120 of file process\_scans.cc.

#### 7.45.1.2 make\_preview()

```
void make_preview (
    Image::ptr orig_image,
    Destination::ptr orig_dest,
    Tags::ptr filetags,
    ImageWriter::ptr preview_file,
    bool can_free = false )
```

Definition at line 47 of file process\_scans.cc.

#### 7.45.1.3 preview\_dir()

```
void preview_dir (
    fs::path dir,
    std::string format,
    std::shared_ptr< Tags > tags )
```

Definition at line 88 of file process\_scans.cc.

## 7.46 sample.h File Reference

### Macros

- #define [SAMPLE](#) float
- #define [SET\\_SAMPLE\\_FORMAT](#)(x) ((x).set\_float())

#### 7.46.1 Macro Definition Documentation

##### 7.46.1.1 SAMPLE

```
#define SAMPLE float
```

Definition at line 23 of file sample.h.

#### 7.46.1.2 SET\_SAMPLE\_FORMAT

```
#define SET_SAMPLE_FORMAT(  
    x ) ((x).set_float())
```

Definition at line 29 of file sample.h.

### 7.47 SOLwriter.cc File Reference

```
#include "ImageFile.hh"  
#include "Image.hh"  
#include "Ditherer.hh"
```

#### Namespaces

- [PhotoFinish](#)

#### Functions

- void [PhotoFinish::write\\_be](#) (void \*ptr, size\_t size, std::ostream &stream)

#### Variables

- unsigned char [PhotoFinish::header](#) [12]

### 7.48 Tags.cc File Reference

```
#include <iostream>  
#include <fstream>  
#include <sstream>  
#include <string>  
#include <map>  
#include <exiv2/exiv2.hpp>  
#include <math.h>  
#include "Image.hh"  
#include "ImageFile.hh"  
#include "Tags.hh"  
#include "Exception.hh"
```

#### Namespaces

- [PhotoFinish](#)

## Functions

- Exiv2::ExifKey [PhotoFinish::exif\\_key\\_read](#) (std::string key\_string)
- Exiv2::Value::AutoPtr [PhotoFinish::exif\\_value\\_read](#) (Exiv2::ExifKey key, std::string value\_string)  
*Read an EXIF value from a string, with optional substitution for enum-style values.*
- Exiv2::IptcKey [PhotoFinish::iptc\\_key\\_read](#) (std::string key\_string)
- Exiv2::XmpKey [PhotoFinish::xmp\\_key\\_read](#) (std::string key\_string)

## 7.49 Tags.hh File Reference

```
#include <exiv2/exiv2.hpp>
#include <boost/filesystem.hpp>
#include <boost/lexical_cast.hpp>
#include <iostream>
#include <string>
#include <map>
#include <memory>
#include <list>
#include "Image.hh"
#include "Destination.hh"
```

## Classes

- class [PhotoFinish::Tags](#)  
*Reads and holds tag information.*

## Namespaces

- [PhotoFinish](#)

## Macros

- #define [StrPair](#)(s, v) std::make\_pair<std::string, std::string>(s, v)

## Typedefs

- typedef std::vector< std::pair< std::string, std::string > > [PhotoFinish::subst\\_table](#)

## Functions

- template<typename Num\_type , typename R\_type >  
Exiv2::ValueType< R\_type > & [PhotoFinish::closest\\_Rational](#) (double value)  
*Find a close rational fraction given a floating-point value.*

## 7.49.1 Macro Definition Documentation

### 7.49.1.1 StrPair

```
#define StrPair(
    s,
    v ) std::make_pair<std::string, std::string>(s, v)
```

Definition at line 38 of file Tags.hh.

## 7.50 Tags\_EXIF\_subst.cc File Reference

```
#include <string>
#include <map>
#include <boost/algorithm/string.hpp>
#include "Tags.hh"
```

### Namespaces

- [PhotoFinish](#)

### Macros

- #define [Key](#)(k, h) std::make\_pair<std::string, subst\_table>(k, h)

### Functions

- Exiv2::ExifKey [PhotoFinish::exif\\_key\\_read](#) (std::string key\_string)
- template<typename Num\_type , typename R\_type >  
Exiv2::Value::AutoPtr [PhotoFinish::parse\\_Rational](#) (std::string s)  
*Parse a string into a rational fraction.*
- Exiv2::Value::AutoPtr [PhotoFinish::exif\\_value\\_read](#) (Exiv2::ExifKey key, std::string value\_string)  
*Read an EXIF value from a string, with optional substitution for enum-style values.*

### Variables

- subst\_table [PhotoFinish::EXIF\\_key\\_subst](#)  
*Map from Image::Exiftool tag names to Exiv2's tag names.*
- std::map< std::string, subst\_table > [PhotoFinish::EXIF\\_value\\_subst](#)

## 7.50.1 Macro Definition Documentation

### 7.50.1.1 Key

```
#define Key(  
    k,  
    h ) std::make_pair<std::string, subst_table>(k, h)
```

Definition at line 91 of file Tags\_EXIF\_subst.cc.

## 7.51 Tags\_IPTC\_subst.cc File Reference

```
#include <string>  
#include <map>  
#include <boost/algorithm/string.hpp>  
#include "Tags.hh"
```

### Namespaces

- [PhotoFinish](#)

### Functions

- Exiv2::IptcKey [PhotoFinish::iptc\\_key\\_read](#) (std::string key\_string)

### Variables

- subst\_table [PhotoFinish::IPTC\\_key\\_subst](#)  
*Map from Image::Exiftool tag names to Exiv2's tag names.*

## 7.52 Tags\_XMP\_subst.cc File Reference

```
#include <string>  
#include <map>  
#include <boost/algorithm/string.hpp>  
#include "Tags.hh"
```

### Namespaces

- [PhotoFinish](#)

### Functions

- Exiv2::XmpKey [PhotoFinish::xmp\\_key\\_read](#) (std::string key\_string)

## Variables

- subst\_table [PhotoFinish::XMP\\_key\\_subst](#)  
*Map from Image::Exiftool tag names to Exiv2's tag names.*

## 7.53 TIFFreader.cc File Reference

```
#include <boost/algorithm/string.hpp>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include <tiffio.h>
#include <tiffio.hxx>
#include "ImageFile.hh"
```

## Namespaces

- [PhotoFinish](#)

## Macros

- #define [TIFFcheck](#)(x) if ((rc = TIFF##x) != 1) throw LibraryError("libtiff", "TIFF" #x " returned " + rc)

### 7.53.1 Macro Definition Documentation

#### 7.53.1.1 TIFFcheck

```
#define TIFFcheck(  
    x ) if ((rc = TIFF##x) != 1) throw LibraryError("libtiff", "TIFF" #x " returned  
" + rc)
```

Definition at line 34 of file TIFFreader.cc.

## 7.54 TIFFwriter.cc File Reference

```
#include <boost/algorithm/string.hpp>
#include <boost/filesystem.hpp>
#include <boost/filesystem/fstream.hpp>
#include <tiffio.h>
#include <tiffio.hxx>
#include "ImageFile.hh"
```

## Namespaces

- [PhotoFinish](#)

## Macros

- `#define TIFFcheck(x) if ((rc = TIFF##x) != 1) throw LibraryError("libtiff", "TIFF" #x " returned " + rc)`

### 7.54.1 Macro Definition Documentation

#### 7.54.1.1 TIFFcheck

```
#define TIFFcheck(  
    x ) if ((rc = TIFF##x) != 1) throw LibraryError("libtiff", "TIFF" #x " returned  
" + rc)
```

Definition at line 34 of file TIFFwriter.cc.

## 7.55 WebP\_ostream.cc File Reference

```
#include <webp/encode.h>  
#include "WebP_ostream.hh"
```

## Namespaces

- [PhotoFinish](#)

## Macros

- `#define min(a, b) ((a) < (b) ? (a) : (b))`

## Functions

- `int PhotoFinish::webp\_stream\_writer\_func (const uint8_t *data, size_t data_size, const WebPPicture *picture)`

*Wrapper around the [webp\\_stream\\_writer](#) class.*

### 7.55.1 Macro Definition Documentation

### 7.55.1.1 min

```
#define min(  
    a,  
    b ) ((a) < (b) ? (a) : (b))
```

Definition at line 133 of file WebP\_ostream.cc.

## 7.56 WebP\_ostream.hh File Reference

```
#include <iostream>  
#include <exiv2/exiv2.hpp>  
#include "CMS.hh"
```

### Classes

- class [PhotoFinish::webp\\_stream\\_writer](#)  
*A custom writer for libwebp that writes using a std::ostream object.*

### Namespaces

- [PhotoFinish](#)

### Functions

- int [PhotoFinish::webp\\_stream\\_writer\\_func](#) (const uint8\_t \*data, size\_t data\_size, const WebPPicture \*picture)  
*Wrapper around the [webp\\_stream\\_writer](#) class.*
- void [PhotoFinish::copy\\_le\\_to](#) (unsigned char \*dest, unsigned int value, unsigned char length)
- unsigned int [PhotoFinish::read\\_le32](#) (const unsigned char \*data)

## 7.57 WebPreader.cc File Reference

```
#include <boost/algorithm/string.hpp>  
#include <webp/decode.h>  
#include <omp.h>  
#include "ImageFile.hh"  
#include "Exception.hh"  
#include "WebP_ostream.hh"
```

### Namespaces

- [PhotoFinish](#)



## 7.58 WebPwriter.cc File Reference

```
#include <boost/algorithm/string.hpp>
#include <webp/encode.h>
#include <omp.h>
#include "ImageFile.hh"
#include "Exception.hh"
#include "WebP_ostream.hh"
```

### Namespaces

- [PhotoFinish](#)

### Variables

- `std::map< std::string, WebPPreset >` [PhotoFinish::WebP\\_presets](#)



# Index

- `__gnu_cxx::new_allocator< Profile >`
    - `CMS::Profile`, [173](#)
  - `__gnu_cxx::new_allocator< Transform >`
    - `CMS::Transform`, [188](#)
  - `_attribute`
    - `PhotoFinish::Uninitialised`, [192](#)
  - `_centrex`
    - `PhotoFinish::Kernel2D`, [155](#)
  - `_centrey`
    - `PhotoFinish::Kernel2D`, [155](#)
  - `_class`
    - `PhotoFinish::NoResults`, [161](#)
    - `PhotoFinish::Unimplemented`, [190](#)
    - `PhotoFinish::Uninitialised`, [192](#)
  - `_defined`
    - `PhotoFinish::Role_Definable`, [176](#)
  - `_destination`
    - `PhotoFinish::NoTargets`, [163](#)
    - `PhotoFinish::PNGReader_cb`, [164](#)
  - `_filepath`
    - `PhotoFinish::FileError`, [100](#)
    - `PhotoFinish::ImageReader`, [138](#)
    - `PhotoFinish::ImageWriter`, [142](#)
  - `_height`
    - `PhotoFinish::D_target`, [65](#)
    - `PhotoFinish::Kernel2D`, [155](#)
  - `_image`
    - `PhotoFinish::PNGReader_cb`, [164](#)
  - `_is_open`
    - `PhotoFinish::ImageReader`, [138](#)
    - `PhotoFinish::ImageWriter`, [142](#)
  - `_method`
    - `PhotoFinish::NoResults`, [161](#)
    - `PhotoFinish::Unimplemented`, [190](#)
  - `_msg`
    - `PhotoFinish::ErrorMsg`, [96](#)
  - `_name`
    - `PhotoFinish::D_target`, [65](#)
  - `_scale`
    - `PhotoFinish::Kernel1Dvar`, [150](#)
  - `_size`
    - `PhotoFinish::D_target`, [65](#)
    - `PhotoFinish::Kernel1Dvar`, [150](#)
  - `_start`
    - `PhotoFinish::Kernel1Dvar`, [150](#)
  - `_to_size`
    - `PhotoFinish::Kernel1Dvar`, [150](#)
  - `_to_size_i`
    - `PhotoFinish::Kernel1Dvar`, [150](#)
  - `_values`
    - `PhotoFinish::Kernel2D`, [155](#)
  - `_weights`
    - `PhotoFinish::Kernel1Dvar`, [150](#)
  - `_width`
    - `PhotoFinish::D_target`, [65](#)
    - `PhotoFinish::Kernel2D`, [155](#)
  - `~D_profile`
    - `PhotoFinish::D_profile`, [56](#)
  - `~Destination`
    - `PhotoFinish::Destination`, [80](#)
  - `~Destinations`
    - `PhotoFinish::Destinations`, [91](#)
  - `~Ditherer`
    - `PhotoFinish::Ditherer`, [94](#)
  - `~Image`
    - `PhotoFinish::Image`, [125](#)
  - `~Kernel1Dvar`
    - `PhotoFinish::Kernel1Dvar`, [146](#)
  - `~Kernel2D`
    - `PhotoFinish::Kernel2D`, [153](#)
  - `~Profile`
    - `CMS::Profile`, [168](#)
  - `~Transform`
    - `CMS::Transform`, [186](#)
  - `~webp_stream_writer`
    - `PhotoFinish::webp_stream_writer`, [196](#)
- `add_exif`
  - `PhotoFinish::webp_stream_writer`, [196](#)
- `add_icc`
  - `PhotoFinish::webp_stream_writer`, [196](#)
- `add_resolution`
  - `PhotoFinish::Tags`, [180](#)
- `add_ruler_pins`
  - `PhotoFinish`, [22](#)
- `add_rulers`
  - `PhotoFinish`, [23](#)
- `add_searchpath`
  - `PhotoFinish::Tags`, [180](#)
- `add_variables`
  - `PhotoFinish::D_JP2`, [42](#)
  - `PhotoFinish::D_JPEG`, [48](#)
  - `PhotoFinish::D_JXR`, [50](#)
  - `PhotoFinish::D_TIFF`, [69](#)
  - `PhotoFinish::D_WebP`, [71](#)
  - `PhotoFinish::Destination`, [80](#)
  - `PhotoFinish::ImageWriter`, [140](#)
- `add_xmp`
  - `PhotoFinish::webp_stream_writer`, [196](#)

- after\_chunk
  - PhotoFinish::webp\_stream\_writer, 196
- alpha\_mult
  - PhotoFinish::Image, 125
- alphaq
  - PhotoFinish::D\_JXR, 51
- artist
  - PhotoFinish::D\_TIFF, 69
- at
  - PhotoFinish::Image, 127
- BYTES\_MASK
  - CMS.cc, 203
- before\_chunk
  - PhotoFinish::webp\_stream\_writer, 197
- begin
  - PhotoFinish::Destinations, 91, 92
- Benchmark.cc, 201
- Benchmark.hh, 201
- benchmark\_mode
  - PhotoFinish, 36
- best\_frame
  - PhotoFinish::Destination, 80
- buffer
  - PhotoFinish::jpeg\_destination\_state\_t, 143
  - PhotoFinish::jpeg\_source\_state\_t, 144
- buffer\_size
  - PhotoFinish::jpeg\_destination\_state\_t, 143
  - PhotoFinish::jpeg\_source\_state\_t, 144
- build
  - PhotoFinish::Kernel1Dvar, 147
- bytes\_per\_channel
  - CMS::Format, 105
- bytes\_per\_pixel
  - CMS::Format, 106
- CHANNELS\_MASK
  - CMS.cc, 203
- CMS.cc, 202
  - BYTES\_MASK, 203
  - CHANNELS\_MASK, 203
  - COLORSPACE\_MASK, 203
  - DOSWAP\_MASK, 203
  - ENDIAN16\_MASK, 203
  - EXTRA\_MASK, 203
  - FLAVOR\_MASK, 204
  - FLOAT\_MASK, 204
  - lcms2\_error\_adaptor, 205
  - lcms2\_errorhandler, 205
  - OPTIMIZED\_MASK, 204
  - PLANAR\_MASK, 204
  - SWAPFIRST\_MASK, 204
- CMS.hh, 205
  - lcms2\_error\_adaptor, 206
- CMS::Format, 102
  - bytes\_per\_channel, 105
  - bytes\_per\_pixel, 106
  - CMYK8, 106
  - channels, 106
  - colour\_model, 106
  - extra\_channels, 106
  - Format, 105
  - Grey16, 107
  - Grey8, 107
  - is\_16bit, 107
  - is\_32bit, 107
  - is\_8bit, 107
  - is\_chocolate, 108
  - is\_double, 108
  - is\_endianswapped, 108
  - is\_float, 108
  - is\_fp, 108
  - is\_half, 109
  - is\_integer, 109
  - is\_optimised, 109
  - is\_packed, 109
  - is\_planar, 109
  - is\_premult\_alpha, 110
  - is\_swapped, 110
  - is\_swappedfirst, 110
  - is\_vanilla, 110
  - LabDouble, 110
  - LabFloat, 111
  - operator cmsUInt32Number, 111
  - RGB16, 111
  - RGB8, 111
  - scaleval, 111
  - set\_16bit, 112
  - set\_32bit, 112
  - set\_8bit, 112
  - set\_channel\_type, 112–114
  - set\_chocolate, 114
  - set\_colour\_model, 114
  - set\_double, 114
  - set\_endianswap, 114
  - set\_extra\_channels, 115
  - set\_float, 115
  - set\_half, 115
  - set\_packed, 115
  - set\_planar, 115
  - set\_premult\_alpha, 116
  - set\_swap, 116
  - set\_swapfirst, 116
  - set\_vanilla, 116
  - total\_channels, 116
  - Transform, 117
  - unset\_endianswap, 117
  - unset\_premult\_alpha, 117
  - unset\_swap, 117
  - unset\_swapfirst, 117
- CMS::Profile, 165
  - \_\_gnu\_cxx::new\_allocator< Profile >, 173
  - ~Profile, 168
  - copyright, 168
  - copyright\_wide, 168
  - description, 169
  - description\_wide, 169

- Lab4, 169
- manufacturer, 169
- manufacturer\_wide, 170
- model, 170
- model\_wide, 170
- operator cmsHPROFILE, 170
- Profile, 167, 168
- ptr, 166
- sGrey, 173
- sRGB, 173
- save\_to\_mem, 171
- set\_copyright, 171
- set\_description, 171, 172
- set\_manufacturer, 172
- set\_model, 172, 173
- CMS::Transform, 185
  - \_\_gnu\_cxx::new\_allocator< Transform >, 188
  - ~Transform, 186
  - change\_formats, 187
  - device\_link, 187
  - input\_format, 187
  - output\_format, 187
  - Proofing, 188
  - ptr, 186
  - Transform, 186
  - transform\_buffer, 188
- CMYK8
  - CMS::Format, 106
- CMS, 11
  - ColourModel, 12
  - Intent, 13
  - istream\_close, 13
  - istream\_read, 13
  - istream\_seek, 14
  - istream\_tell, 14
  - istream\_write, 14
  - OpenIOHandlerFromIFStream, 14
  - OpenIOHandlerFromIStream, 14
  - operator<<, 15
  - ostream\_close, 15
  - ostream\_read, 15
  - ostream\_seek, 15
  - ostream\_tell, 16
  - ostream\_write, 16
- COLORSPACE\_MASK
  - CMS.cc, 203
- change\_formats
  - CMS::Transform, 187
- channels
  - CMS::Format, 106
- check\_rowdata\_alloc
  - PhotoFinish::Image, 127
- clear\_profile
  - PhotoFinish::Destination, 80
- closest\_Rational
  - PhotoFinish, 23
- cmsBaseType
  - PhotoFinish::Ditherer, 95
- cmsTypeError
  - PhotoFinish::cmsTypeError, 39
- colour\_model
  - CMS::Format, 106
- ColourModel
  - CMS, 12
- compression
  - PhotoFinish::D\_TIFF, 69
- const\_iterator
  - PhotoFinish::Destinations, 90
- convolve
  - PhotoFinish::Kernel2D, 153
- convolve\_h
  - PhotoFinish::Kernel1Dvar, 147
- convolve\_h\_type
  - PhotoFinish::Kernel1Dvar, 147
- convolve\_h\_type\_channels
  - PhotoFinish::Kernel1Dvar, 148
- convolve\_type
  - PhotoFinish::Kernel2D, 154
- convolve\_type\_channels
  - PhotoFinish::Kernel2D, 154
- convolve\_v
  - PhotoFinish::Kernel1Dvar, 148
- convolve\_v\_type
  - PhotoFinish::Kernel1Dvar, 148
- convolve\_v\_type\_channels
  - PhotoFinish::Kernel1Dvar, 149
- copy\_from
  - PhotoFinish::Tags, 181
- copy\_le\_to
  - PhotoFinish, 23
- copy\_to
  - PhotoFinish::Tags, 181
- copyright
  - CMS::Profile, 168
  - PhotoFinish::D\_TIFF, 69
- copyright\_wide
  - CMS::Profile, 168
- count
  - PhotoFinish::Destinations, 91
- create
  - PhotoFinish::Kernel1Dvar, 149
  - PhotoFinish::Kernel2D, 154
- crop\_h
  - PhotoFinish::Frame, 120
- crop\_resize
  - PhotoFinish::Frame, 120
- crop\_w
  - PhotoFinish::Frame, 121
- crop\_x
  - PhotoFinish::Frame, 121
- crop\_y
  - PhotoFinish::Frame, 121
- CropSolution.cc, 207
  - max, 207
  - min, 207
  - sqr, 208

- CropSolution.hh, 208
- CropSolver
  - PhotoFinish::CropSolver, 40
- D\_JP2
  - PhotoFinish::D\_JP2, 42
- D\_JPEG
  - PhotoFinish::D\_JPEG, 47
- D\_JXR
  - PhotoFinish::D\_JXR, 50
- D\_PNG
  - PhotoFinish::D\_PNG, 54
- D\_TIFF
  - PhotoFinish::D\_TIFF, 68
- D\_WebP
  - PhotoFinish::D\_WebP, 71
- D\_profile
  - PhotoFinish::D\_profile, 56
- D\_resize
  - PhotoFinish::D\_resize, 59
- D\_sharpen
  - PhotoFinish::D\_sharpen, 61
- D\_target
  - PhotoFinish::D\_target, 63
- D\_thumbnail
  - PhotoFinish::D\_thumbnail, 66
- DOSWAP\_MASK
  - CMS.cc, 203
- data
  - PhotoFinish::D\_profile, 57
- data\_size
  - PhotoFinish::D\_profile, 57
- default\_profile
  - PhotoFinish::Image, 127
- definable
  - PhotoFinish::definable, 74, 75
- Definable.hh, 209
- defined
  - PhotoFinish::Role\_Definable, 175, 176
  - PhotoFinish::definable, 75
- depth
  - PhotoFinish::Destination, 81
- description
  - CMS::Profile, 169
- description\_wide
  - CMS::Profile, 169
- Destination
  - PhotoFinish::Destination, 79
- Destination.cc, 209
- Destination.hh, 209
- Destination\_items.cc, 210
- Destination\_items.hh, 210
- DestinationError
  - PhotoFinish::DestinationError, 88
- Destinations
  - PhotoFinish::Destinations, 90
- device\_link
  - CMS::Transform, 187
- dir
  - PhotoFinish::Destination, 81
- dither
  - PhotoFinish::Ditherer, 94
- Ditherer
  - PhotoFinish::Ditherer, 94
- Ditherer.cc, 211
  - nextpos, 212
  - pos, 212
  - prevpos, 212
- Ditherer.hh, 212
- dupe
  - PhotoFinish::Destination, 81
  - PhotoFinish::Tags, 181
- ENDIAN16\_MASK
  - CMS.cc, 203
- EXIF\_key\_subst
  - PhotoFinish, 36
- EXIF\_value\_subst
  - PhotoFinish, 37
- EXIFtags
  - PhotoFinish::Image, 128
  - PhotoFinish::Tags, 181
- EXTRA\_MASK
  - CMS.cc, 203
- elapsed
  - PhotoFinish::Timer, 184
- elapsed\_ns
  - PhotoFinish::Timer, 184
- embed\_tags
  - PhotoFinish::ImageWriter, 141
- end
  - PhotoFinish::Destinations, 91–93
  - PhotoFinish::PNGreader\_cb, 164
- error\_callback
  - PhotoFinish, 23
- ErrorMsg
  - PhotoFinish::ErrorMsg, 96
- eval
  - PhotoFinish::Kernel1Dvar, 149
- Exception.hh, 213
- exif\_key\_read
  - PhotoFinish, 24
- exif\_value\_read
  - PhotoFinish, 24
- exists
  - PhotoFinish, 24
- extra\_channels
  - CMS::Format, 106
- extract\_tags
  - PhotoFinish::ImageReader, 137
- FLAVOR\_MASK
  - CMS.cc, 204
- FLOAT\_MASK
  - CMS.cc, 204
- FileContentError
  - PhotoFinish::FileContentError, 97, 98
- FileError

- PhotoFinish::FileError, [99](#), [100](#)
- FileOpenError
  - PhotoFinish::FileOpenError, [101](#), [102](#)
- filepath
  - PhotoFinish::D\_profile, [57](#)
  - PhotoFinish::ImageFilepath, [134](#)
- filter
  - PhotoFinish::D\_resize, [60](#)
- fix\_filepath
  - PhotoFinish::ImageFilepath, [135](#)
- fixed\_filepath
  - PhotoFinish::ImageFilepath, [135](#)
- FmtPair
  - JXR.hh, [222](#)
- forcegrey
  - PhotoFinish::Destination, [81](#)
- forcergb
  - PhotoFinish::Destination, [81](#)
- Format
  - CMS::Format, [105](#)
- format
  - PhotoFinish::Destination, [82](#)
  - PhotoFinish::Image, [128](#)
  - PhotoFinish::ImageFilepath, [135](#)
- Frame
  - PhotoFinish::Frame, [119](#)
- Frame.cc, [214](#)
- Frame.hh, [214](#)
- free\_row
  - PhotoFinish::Image, [128](#)
- GaussianSharpen
  - PhotoFinish::GaussianSharpen, [122](#)
- generate
  - PhotoFinish::D\_thumbnail, [66](#)
- get
  - PhotoFinish::definable, [75](#)
- get\_profile
  - PhotoFinish::Destination, [82](#)
- Grey16
  - CMS::Format, [107](#)
- Grey8
  - CMS::Format, [107](#)
- has\_data
  - PhotoFinish::D\_profile, [57](#)
- has\_profile
  - PhotoFinish::Image, [128](#)
- has\_targets
  - PhotoFinish::Destination, [82](#)
- hash
  - PhotoFinish, [21](#)
- header
  - PhotoFinish, [37](#)
- height
  - PhotoFinish::D\_target, [64](#)
  - PhotoFinish::Image, [128](#)
- IPTC\_key\_subst
  - PhotoFinish, [37](#)
- IPTCtags
  - PhotoFinish::Image, [129](#)
  - PhotoFinish::Tags, [181](#)
- Image
  - PhotoFinish::Image, [125](#)
- Image.cc, [214](#)
- Image.hh, [215](#)
- ImageFile.cc, [216](#)
- ImageFile.hh, [216](#)
- ImageFilepath
  - PhotoFinish::ImageFilepath, [134](#)
- ImageReader
  - PhotoFinish::ImageReader, [137](#)
- ImageWriter
  - PhotoFinish::ImageWriter, [140](#)
- info
  - PhotoFinish::PNGReader\_cb, [164](#)
- info\_callback
  - PhotoFinish, [24](#)
- input\_format
  - CMS::Transform, [187](#)
- Intent
  - CMS, [13](#)
- intent
  - PhotoFinish::Destination, [82](#)
- iptc\_key\_read
  - PhotoFinish, [24](#)
- is
  - PhotoFinish::jpeg\_source\_state\_t, [144](#)
- is\_16bit
  - CMS::Format, [107](#)
- is\_32bit
  - CMS::Format, [107](#)
- is\_8bit
  - CMS::Format, [107](#)
- is\_chocolate
  - CMS::Format, [108](#)
- is\_double
  - CMS::Format, [108](#)
- is\_endianswapped
  - CMS::Format, [108](#)
- is\_float
  - CMS::Format, [108](#)
- is\_fp
  - CMS::Format, [108](#)
- is\_half
  - CMS::Format, [109](#)
- is\_integer
  - CMS::Format, [109](#)
- is\_optimised
  - CMS::Format, [109](#)
- is\_packed
  - CMS::Format, [109](#)
- is\_planar
  - CMS::Format, [109](#)
- is\_premult\_alpha
  - CMS::Format, [110](#)

- is\_swapped
  - CMS::Format, [110](#)
- is\_swappedfirst
  - CMS::Format, [110](#)
- is\_vanilla
  - CMS::Format, [110](#)
- istream\_close
  - CMS, [13](#)
- istream\_read
  - CMS, [13](#)
- istream\_seek
  - CMS, [14](#)
- istream\_tell
  - CMS, [14](#)
- istream\_write
  - CMS, [14](#)
- iterator
  - PhotoFinish::Destinations, [90](#)
- JP2.hh, [217](#)
- JP2\_callbacks.cc, [217](#)
- JP2reader.cc, [218](#)
- JP2writer.cc, [218](#)
- JPEG.hh, [218](#)
- JPEG\_istream.cc, [219](#)
- JPEG\_profiles.cc, [220](#)
- JPEG\_scans.cc, [221](#)
- JPEGreader.cc, [221](#)
- JPEGwriter.cc, [221](#)
- JXR.hh, [222](#)
  - FmtPair, [222](#)
  - JXRcheck, [222](#)
- JXR\_format\_table
  - PhotoFinish, [37](#)
- JXR\_formats.cc, [223](#)
- JXRcheck
  - JXR.hh, [222](#)
- JXRreader.cc, [223](#)
  - jxr\_metadata\_data, [224](#)
  - jxr\_metadata\_size, [224](#)
- JXRwriter.cc, [224](#)
- jp2
  - PhotoFinish::Destination, [82](#)
- jpeg
  - PhotoFinish::Destination, [83](#)
- jpeg\_error\_exit
  - PhotoFinish, [25](#)
- jpeg\_istream\_fill\_input\_buffer
  - PhotoFinish, [25](#)
- jpeg\_istream\_init\_source
  - PhotoFinish, [25](#)
- jpeg\_istream\_resync\_to\_restart
  - PhotoFinish, [25](#)
- jpeg\_istream\_skip\_input\_data
  - PhotoFinish, [25](#)
- jpeg\_istream\_src
  - PhotoFinish, [26](#)
- jpeg\_istream\_src\_free
  - PhotoFinish, [26](#)
- jpeg\_istream\_term\_source
  - PhotoFinish, [26](#)
- jpeg\_ostream\_dest
  - PhotoFinish, [26](#)
- jpeg\_ostream\_dest\_free
  - PhotoFinish, [27](#)
- jpeg\_read\_profile
  - PhotoFinish, [27](#)
- jpeg\_write\_profile
  - PhotoFinish, [27](#)
- jpegfile\_scan\_RGB
  - PhotoFinish, [28](#)
- jpegfile\_scan\_greyscale
  - PhotoFinish, [27](#)
- jxr
  - PhotoFinish::Destination, [83](#)
- jxr\_cms\_format
  - PhotoFinish, [28](#)
- jxr\_format\_subst
  - PhotoFinish, [21](#)
- jxr\_metadata\_data
  - JXRreader.cc, [224](#)
- jxr\_metadata\_size
  - JXRreader.cc, [224](#)
- jxr\_pixel\_format
  - PhotoFinish, [28](#)
- Kernel1Dvar
  - PhotoFinish::Kernel1Dvar, [146](#)
- Kernel1Dvar.cc, [225](#)
  - min, [225](#)
  - sqr, [225](#)
- Kernel1Dvar.hh, [225](#)
- Kernel2D.cc, [226](#)
  - sqr, [226](#)
- Kernel2D.hh, [226](#)
- Kernel2D
  - PhotoFinish::Kernel2D, [152](#), [153](#)
- Key
  - Tags\_EXIF\_subst.cc, [234](#)
- LCMS2ErrorHandler.cc, [227](#)
- Lab4
  - CMS::Profile, [169](#)
- LabDouble
  - CMS::Format, [110](#)
- LabFloat
  - CMS::Format, [111](#)
- Lanczos
  - PhotoFinish::Lanczos, [156](#)
- lanczos
  - PhotoFinish::D\_resize, [60](#)
- last\_write\_time
  - PhotoFinish, [28](#)
- lcms2\_error\_adaptor
  - CMS.cc, [205](#)
  - CMS.hh, [206](#)
  - PhotoFinish, [28](#)
- lcms2\_errorhandler



- CMS.cc, [205](#)
- PhotoFinish, [29](#)
- LibraryError
  - PhotoFinish::LibraryError, [158](#)
- limitval
  - PhotoFinish, [29](#)
- limitval< double >
  - PhotoFinish, [29](#)
- limitval< float >
  - PhotoFinish, [29](#)
- limitval< unsigned char >
  - PhotoFinish, [29](#)
- limitval< unsigned int >
  - PhotoFinish, [30](#)
- limitval< unsigned long long >
  - PhotoFinish, [30](#)
- limitval< unsigned short int >
  - PhotoFinish, [30](#)
- Load
  - PhotoFinish::Destinations, [92](#)
- load
  - PhotoFinish::Tags, [182](#)
- lossless
  - PhotoFinish::D\_WebP, [71](#)
- lossy
  - PhotoFinish::D\_WebP, [72](#)
- main
  - photofinish.cc, [228](#)
  - process\_scans.cc, [230](#)
- make\_preview
  - process\_scans.cc, [231](#)
- make\_thumbnail
  - PhotoFinish::Tags, [182](#)
- manufacturer
  - CMS::Profile, [169](#)
- manufacturer\_wide
  - CMS::Profile, [170](#)
- max
  - CropSolution.cc, [207](#)
- maxheight
  - PhotoFinish::D\_thumbnail, [67](#)
- maxwidth
  - PhotoFinish::D\_thumbnail, [67](#)
- MemAllocError
  - PhotoFinish::MemAllocError, [159](#)
- method
  - PhotoFinish::D\_WebP, [72](#)
- min
  - CropSolution.cc, [207](#)
  - Kernel1Dvar.cc, [225](#)
  - WebP\_ostream.cc, [237](#)
- model
  - CMS::Profile, [170](#)
- model\_wide
  - CMS::Profile, [170](#)
- modify\_chunk
  - PhotoFinish::webp\_stream\_writer, [197](#)
- modify\_format
  - PhotoFinish::Destination, [83](#)
- modify\_vp8x
  - PhotoFinish::webp\_stream\_writer, [197](#)
- multihash
  - PhotoFinish, [21](#)
- name
  - PhotoFinish::D\_profile, [58](#)
  - PhotoFinish::D\_target, [64](#)
  - PhotoFinish::Destination, [83](#)
- nextpos
  - Ditherer.cc, [212](#)
- NoResults
  - PhotoFinish::NoResults, [160](#)
- NoTargets
  - PhotoFinish::NoTargets, [162](#)
- noresize
  - PhotoFinish::Destination, [83](#)
- num\_qualities
  - PhotoFinish::D\_JP2, [43](#)
- num\_rates
  - PhotoFinish::D\_JP2, [43](#)
- num\_targets
  - PhotoFinish::Destination, [84](#)
- numresolutions
  - PhotoFinish::D\_JP2, [43](#)
- OPTIMIZED\_MASK
  - CMS.cc, [204](#)
- open
  - PhotoFinish::ImageReader, [137](#)
  - PhotoFinish::ImageWriter, [141](#)
- OpenIOHandlerFromIFStream
  - CMS, [14](#)
- OpenIOHandlerFromIStream
  - CMS, [14](#)
- operator cmsHPROFILE
  - CMS::Profile, [170](#)
- operator cmsUInt32Number
  - CMS::Format, [111](#)
- operator T
  - PhotoFinish::definable, [76](#)
- operator<<
  - CMS, [15](#)
  - PhotoFinish, [30](#)
  - PhotoFinish::ImageFilepath, [135](#)
  - PhotoFinish::definable, [77](#)
- operator->
  - PhotoFinish::definable, [76](#)
- operator=
  - PhotoFinish::D\_profile, [58](#)
  - PhotoFinish::Destination, [84](#)
  - PhotoFinish::Destinations, [92](#)
  - PhotoFinish::definable, [76](#)
- operator[]
  - PhotoFinish::Destinations, [92](#)
- os
  - PhotoFinish::jpeg\_destination\_state\_t, [143](#)
- ostream\_close

- CMS, 15
- ostream\_read
  - CMS, 15
- ostream\_seek
  - CMS, 15
- ostream\_tell
  - CMS, 16
- ostream\_write
  - CMS, 16
- output\_format
  - CMS::Transform, 187
- overlap
  - PhotoFinish::D\_JXR, 51
- PLANAR\_MASK
  - CMS.cc, 204
- PNGreader.cc, 228
- PNGreader\_cb
  - PhotoFinish::PNGreader\_cb, 163
- PNGreader\_cb.cc, 228
- PNGreader\_cb.hh, 229
- PNGwriter.cc, 229
- parse\_Rational
  - PhotoFinish, 30
- PhotoFinish, 16
  - add\_ruler\_pins, 22
  - add\_rulers, 23
  - benchmark\_mode, 36
  - closest\_Rational, 23
  - copy\_le\_to, 23
  - EXIF\_key\_subst, 36
  - EXIF\_value\_subst, 37
  - error\_callback, 23
  - exif\_key\_read, 24
  - exif\_value\_read, 24
  - exists, 24
  - hash, 21
  - header, 37
  - IPTC\_key\_subst, 37
  - info\_callback, 24
  - iptc\_key\_read, 24
  - JXR\_format\_table, 37
  - jpeg\_error\_exit, 25
  - jpeg\_istream\_fill\_input\_buffer, 25
  - jpeg\_istream\_init\_source, 25
  - jpeg\_istream\_resync\_to\_restart, 25
  - jpeg\_istream\_skip\_input\_data, 25
  - jpeg\_istream\_src, 26
  - jpeg\_istream\_src\_free, 26
  - jpeg\_istream\_term\_source, 26
  - jpeg\_ostream\_dest, 26
  - jpeg\_ostream\_dest\_free, 27
  - jpeg\_read\_profile, 27
  - jpeg\_write\_profile, 27
  - jpegfile\_scan\_RGB, 28
  - jpegfile\_scan\_greyscale, 27
  - jxr\_cms\_format, 28
  - jxr\_format\_subst, 21
  - jxr\_pixel\_format, 28
  - last\_write\_time, 28
  - lcms2\_error\_adaptor, 28
  - lcms2\_errorhandler, 29
  - limitval, 29
  - limitval< double >, 29
  - limitval< float >, 29
  - limitval< unsigned char >, 29
  - limitval< unsigned int >, 30
  - limitval< unsigned long long >, 30
  - limitval< unsigned short int >, 30
  - multihash, 21
  - operator<<, 30
  - parse\_Rational, 30
  - png\_end\_cb, 31
  - png\_flush\_ostream\_cb, 31
  - png\_info\_cb, 31
  - png\_row\_cb, 31
  - png\_write\_ostream\_cb, 32
  - profile\_name, 32
  - read\_le32, 32
  - read\_planar, 32
  - rulerlist, 22
  - rulerpair, 22
  - scaleval, 33
  - scaleval< double >, 33
  - scaleval< float >, 33
  - scaleval< unsigned char >, 33
  - scaleval< unsigned int >, 33
  - scaleval< unsigned long long >, 34
  - scaleval< unsigned short int >, 34
  - stringlist, 22
  - subst\_table, 22
  - transfer\_alpha, 34
  - transfer\_alpha\_typed, 34
  - transfer\_alpha\_typed2, 34
  - warning\_callback, 35
  - WebP\_presets, 38
  - webp\_stream\_writer\_func, 35
  - write\_be, 35
  - write\_packed, 35
  - write\_planar, 36
  - XMP\_key\_subst, 38
  - xmp\_key\_read, 36
  - PhotoFinish::CropSolver, 40
    - CropSolver, 40
    - solve, 41
  - PhotoFinish::D\_JP2, 41
    - add\_variables, 42
    - D\_JP2, 42
    - num\_qualities, 43
    - num\_rates, 43
    - numresolutions, 43
    - prog\_order, 43
    - quality, 43
    - rate, 44
    - read\_config, 44
    - reversible, 44
    - set\_irreversible, 44

- set\_numresolutions, 44
- set\_prog\_order, 45
- set\_qualities, 45
- set\_quality, 45
- set\_rate, 45
- set\_rates, 45
- set\_reversible, 46
- set\_tile\_size, 46
- tile\_size, 46
- PhotoFinish::D\_JPEG, 46
  - add\_variables, 48
  - D\_JPEG, 47
  - progressive, 48
  - quality, 48
  - read\_config, 48
  - sample, 48
  - set\_progressive, 49
  - set\_quality, 49
  - set\_sample, 49
- PhotoFinish::D\_JXR, 49
  - add\_variables, 50
  - alphaq, 51
  - D\_JXR, 50
  - overlap, 51
  - progressive, 51
  - quality, 51
  - read\_config, 51
  - set\_alphaq, 52
  - set\_overlap, 52
  - set\_progressive, 52
  - set\_quality, 52
  - set\_sequential, 52
  - set\_subsampling, 52
  - set\_tilsize, 53
  - subsampling, 53
  - tilsize, 53
- PhotoFinish::D\_PNG, 53
  - D\_PNG, 54
  - read\_config, 54
- PhotoFinish::D\_TIFF, 67
  - add\_variables, 69
  - artist, 69
  - compression, 69
  - copyright, 69
  - D\_TIFF, 68
  - read\_config, 69
  - set\_artist, 70
  - set\_compression, 70
  - set\_copyright, 70
- PhotoFinish::D\_WebP, 70
  - add\_variables, 71
  - D\_WebP, 71
  - lossless, 71
  - lossy, 72
  - method, 72
  - preset, 72
  - quality, 72
  - read\_config, 72
  - set\_lossless, 72
  - set\_lossy, 73
  - set\_method, 73
  - set\_preset, 73
  - set\_quality, 73
- PhotoFinish::D\_profile, 54
  - ~D\_profile, 56
  - D\_profile, 56
  - data, 57
  - data\_size, 57
  - filepath, 57
  - has\_data, 57
  - name, 58
  - operator=, 58
  - profile, 58
  - ptr, 55
  - read\_config, 58
- PhotoFinish::D\_resize, 59
  - D\_resize, 59
  - filter, 60
  - lanczos, 60
  - read\_config, 60
  - support, 60
- PhotoFinish::D\_sharpen, 61
  - D\_sharpen, 61
  - radius, 61
  - read\_config, 62
  - sigma, 62
- PhotoFinish::D\_target, 62
  - \_height, 65
  - \_name, 65
  - \_size, 65
  - \_width, 65
  - D\_target, 63
  - height, 64
  - name, 64
  - ptr, 63
  - read\_config, 64
  - size, 64
  - width, 64
- PhotoFinish::D\_thumbnail, 66
  - D\_thumbnail, 66
  - generate, 66
  - maxheight, 67
  - maxwidth, 67
  - read\_config, 67
- PhotoFinish::Destination, 78
  - ~Destination, 80
  - add\_variables, 80
  - best\_frame, 80
  - clear\_profile, 80
  - depth, 81
  - Destination, 79
  - dir, 81
  - dupe, 81
  - forcegrey, 81
  - forcergb, 81
  - format, 82

- get\_profile, 82
- has\_targets, 82
- intent, 82
- jp2, 82
- jpeg, 83
- jxr, 83
- modify\_format, 83
- name, 83
- noresize, 83
- num\_targets, 84
- operator=, 84
- png, 84
- profile, 84
- ptr, 79
- read\_config, 84
- resize, 85
- set\_depth, 85
- set\_jp2, 85
- set\_jpeg, 85
- set\_jxr, 85
- set\_png, 86
- set\_profile, 86
- set\_tiff, 86
- set\_webp, 86
- sharpen, 87
- size, 87
- targets, 87
- thumbnail, 87
- tiff, 87
- webp, 87
- PhotoFinish::DestinationError, 88
  - DestinationError, 88
  - what, 89
- PhotoFinish::Destinations, 89
  - ~Destinations, 91
  - begin, 91, 92
  - const\_iterator, 90
  - count, 91
  - Destinations, 90
  - end, 91–93
  - iterator, 90
  - Load, 92
  - operator=, 92
  - operator[], 92
- PhotoFinish::Ditherer, 93
  - ~Ditherer, 94
  - cmsBaseType, 95
  - dither, 94
  - Ditherer, 94
- PhotoFinish::ErrorMsg, 95
  - \_msg, 96
  - ErrorMsg, 96
  - what, 96
- PhotoFinish::FileContentError, 97
  - FileContentError, 97, 98
  - what, 98
- PhotoFinish::FileError, 98
  - \_filepath, 100
  - FileError, 99, 100
  - what, 100
- PhotoFinish::FileOpenError, 101
  - FileOpenError, 101, 102
  - what, 102
- PhotoFinish::Frame, 118
  - crop\_h, 120
  - crop\_resize, 120
  - crop\_w, 121
  - crop\_x, 121
  - crop\_y, 121
  - Frame, 119
  - ptr, 119
  - waste, 121
- PhotoFinish::GaussianSharpen, 122
  - GaussianSharpen, 122
- PhotoFinish::Image, 123
  - ~Image, 125
  - alpha\_mult, 125
  - at, 127
  - check\_rowdata\_alloc, 127
  - default\_profile, 127
  - EXIFtags, 128
  - format, 128
  - free\_row, 128
  - has\_profile, 128
  - height, 128
  - IPTCtags, 129
  - Image, 125
  - pixel\_size, 129
  - profile, 129
  - ptr, 125
  - row, 129
  - row\_size, 129
  - set\_profile, 130
  - set\_resolution, 130
  - set\_resolution\_from\_size, 130
  - set\_xres, 130
  - set\_yres, 131
  - transform\_colour, 131
  - transform\_colour\_inplace, 131
  - un\_alpha\_mult, 132
  - width, 132
  - XMPtags, 132
  - xres, 132
  - yres, 133
- PhotoFinish::ImageFilepath, 133
  - filepath, 134
  - fix\_filepath, 135
  - fixed\_filepath, 135
  - format, 135
  - ImageFilepath, 134
  - operator<<, 135
- PhotoFinish::ImageReader, 136
  - \_filepath, 138
  - \_is\_open, 138
  - extract\_tags, 137
  - ImageReader, 137

- open, 137
- ptr, 136
- read, 138
- PhotoFinish::ImageWriter, 139
  - \_filepath, 142
  - \_is\_open, 142
  - add\_variables, 140
  - embed\_tags, 141
  - ImageWriter, 140
  - open, 141
  - preferred\_format, 141
  - ptr, 140
  - write, 141
- PhotoFinish::Kernel1Dvar, 145
  - \_scale, 150
  - \_size, 150
  - \_start, 150
  - \_to\_size, 150
  - \_to\_size\_i, 150
  - \_weights, 150
  - ~Kernel1Dvar, 146
  - build, 147
  - convolve\_h, 147
  - convolve\_h\_type, 147
  - convolve\_h\_type\_channels, 148
  - convolve\_v, 148
  - convolve\_v\_type, 148
  - convolve\_v\_type\_channels, 149
  - create, 149
  - eval, 149
  - Kernel1Dvar, 146
  - ptr, 146
  - range, 149
- PhotoFinish::Kernel2D, 151
  - \_centrex, 155
  - \_centrey, 155
  - \_height, 155
  - \_values, 155
  - \_width, 155
  - ~Kernel2D, 153
  - convolve, 153
  - convolve\_type, 154
  - convolve\_type\_channels, 154
  - create, 154
  - Kernel2D, 152, 153
  - ptr, 152
- PhotoFinish::Lanczos, 156
  - Lanczos, 156
- PhotoFinish::LibraryError, 157
  - LibraryError, 158
  - what, 158
- PhotoFinish::MemAllocError, 158
  - MemAllocError, 159
  - what, 159
- PhotoFinish::NoResults, 160
  - \_class, 161
  - \_method, 161
  - NoResults, 160
- what, 161
- PhotoFinish::NoTargets, 161
  - \_destination, 163
  - NoTargets, 162
  - what, 162
- PhotoFinish::PNGReader\_cb, 163
  - \_destination, 164
  - \_image, 164
  - end, 164
  - info, 164
  - PNGReader\_cb, 163
  - row, 164
- PhotoFinish::Role\_Definable, 174
  - \_defined, 176
  - defined, 175, 176
  - Role\_Definable, 175
  - set\_defined, 175
  - undefine, 176
- PhotoFinish::SOLwriter, 177
  - preferred\_format, 177
  - SOLwriter, 177
  - write, 178
- PhotoFinish::Tags, 178
  - add\_resolution, 180
  - add\_searchpath, 180
  - copy\_from, 181
  - copy\_to, 181
  - dupe, 181
  - EXIFtags, 181
  - IPTCtags, 181
  - load, 182
  - make\_thumbnail, 182
  - ptr, 179
  - Tags, 180
  - try\_load, 182
  - variables, 182
  - XMPtags, 183
- PhotoFinish::Timer, 183
  - elapsed, 184
  - elapsed\_ns, 184
  - start, 184
  - stop, 184
  - Timer, 184
- PhotoFinish::Unimplemented, 189
  - \_class, 190
  - \_method, 190
  - Unimplemented, 189
  - what, 190
- PhotoFinish::Uninitialised, 190
  - \_attribute, 192
  - \_class, 192
  - Uninitialised, 191
  - what, 192
- PhotoFinish::UnknownFileType, 193
  - UnknownFileType, 193, 194
  - what, 194
- PhotoFinish::WebPError, 198
  - WebPError, 198

- what, 199
- PhotoFinish::cmsTypeError, 39
  - cmsTypeError, 39
  - what, 40
- PhotoFinish::definable
  - definable, 74, 75
  - defined, 75
  - get, 75
  - operator T, 76
  - operator<<, 77
  - operator>, 76
  - operator=, 76
  - set\_defined, 77
  - undefine, 77
- PhotoFinish::definable< T >, 74
- PhotoFinish::jpeg\_destination\_state\_t, 142
  - buffer, 143
  - buffer\_size, 143
  - os, 143
- PhotoFinish::jpeg\_source\_state\_t, 143
  - buffer, 144
  - buffer\_size, 144
  - is, 144
- PhotoFinish::webp\_stream\_writer, 194
  - ~webp\_stream\_writer, 196
  - add\_exif, 196
  - add\_icc, 196
  - add\_xmp, 196
  - after\_chunk, 196
  - before\_chunk, 197
  - modify\_chunk, 197
  - modify\_vp8x, 197
  - webp\_stream\_writer, 195
  - write, 197
  - write\_chunk, 197
- photofinish.cc, 227
  - main, 228
- pixel\_size
  - PhotoFinish::Image, 129
- png
  - PhotoFinish::Destination, 84
- png\_end\_cb
  - PhotoFinish, 31
- png\_flush\_ostream\_cb
  - PhotoFinish, 31
- png\_info\_cb
  - PhotoFinish, 31
- png\_row\_cb
  - PhotoFinish, 31
- png\_write\_ostream\_cb
  - PhotoFinish, 32
- pos
  - Ditherer.cc, 212
- preferred\_format
  - PhotoFinish::ImageWriter, 141
  - PhotoFinish::SOLwriter, 177
- preset
  - PhotoFinish::D\_WebP, 72
- preview\_dir
  - process\_scans.cc, 231
- prevpos
  - Ditherer.cc, 212
- process\_scans.cc, 230
  - main, 230
  - make\_preview, 231
  - preview\_dir, 231
- Profile
  - CMS::Profile, 167, 168
- profile
  - PhotoFinish::D\_profile, 58
  - PhotoFinish::Destination, 84
  - PhotoFinish::Image, 129
- profile\_name
  - PhotoFinish, 32
- prog\_order
  - PhotoFinish::D\_JP2, 43
- progressive
  - PhotoFinish::D\_JPEG, 48
  - PhotoFinish::D\_JXR, 51
- Proofing
  - CMS::Transform, 188
- ptr
  - CMS::Profile, 166
  - CMS::Transform, 186
  - PhotoFinish::D\_profile, 55
  - PhotoFinish::D\_target, 63
  - PhotoFinish::Destination, 79
  - PhotoFinish::Frame, 119
  - PhotoFinish::Image, 125
  - PhotoFinish::ImageReader, 136
  - PhotoFinish::ImageWriter, 140
  - PhotoFinish::Kernel1Dvar, 146
  - PhotoFinish::Kernel2D, 152
  - PhotoFinish::Tags, 179
- quality
  - PhotoFinish::D\_JP2, 43
  - PhotoFinish::D\_JPEG, 48
  - PhotoFinish::D\_JXR, 51
  - PhotoFinish::D\_WebP, 72
- RGB16
  - CMS::Format, 111
- RGB8
  - CMS::Format, 111
- radius
  - PhotoFinish::D\_sharpen, 61
- range
  - PhotoFinish::Kernel1Dvar, 149
- rate
  - PhotoFinish::D\_JP2, 44
- read
  - PhotoFinish::ImageReader, 138
- read\_config
  - PhotoFinish::D\_JP2, 44
  - PhotoFinish::D\_JPEG, 48
  - PhotoFinish::D\_JXR, 51

- PhotoFinish::D\_PNG, [54](#)
- PhotoFinish::D\_TIFF, [69](#)
- PhotoFinish::D\_WebP, [72](#)
- PhotoFinish::D\_profile, [58](#)
- PhotoFinish::D\_resize, [60](#)
- PhotoFinish::D\_sharpen, [62](#)
- PhotoFinish::D\_target, [64](#)
- PhotoFinish::D\_thumbnail, [67](#)
- PhotoFinish::Destination, [84](#)
- read\_le32
  - PhotoFinish, [32](#)
- read\_planar
  - PhotoFinish, [32](#)
- resize
  - PhotoFinish::Destination, [85](#)
- reversible
  - PhotoFinish::D\_JP2, [44](#)
- Role\_Definable
  - PhotoFinish::Role\_Definable, [175](#)
- row
  - PhotoFinish::Image, [129](#)
  - PhotoFinish::PNGreader\_cb, [164](#)
- row\_size
  - PhotoFinish::Image, [129](#)
- rulerlist
  - PhotoFinish, [22](#)
- rulerpair
  - PhotoFinish, [22](#)
- SAMPLE
  - sample.h, [231](#)
- SET\_SAMPLE\_FORMAT
  - sample.h, [231](#)
- sGrey
  - CMS::Profile, [173](#)
- SOLwriter
  - PhotoFinish::SOLwriter, [177](#)
- SOLwriter.cc, [232](#)
- sRGB
  - CMS::Profile, [173](#)
- SWAPFIRST\_MASK
  - CMS.cc, [204](#)
- sample
  - PhotoFinish::D\_JPEG, [48](#)
- sample.h, [231](#)
  - SAMPLE, [231](#)
  - SET\_SAMPLE\_FORMAT, [231](#)
- save\_to\_mem
  - CMS::Profile, [171](#)
- scaleval
  - CMS::Format, [111](#)
  - PhotoFinish, [33](#)
- scaleval< double >
  - PhotoFinish, [33](#)
- scaleval< float >
  - PhotoFinish, [33](#)
- scaleval< unsigned char >
  - PhotoFinish, [33](#)
- scaleval< unsigned int >
  - PhotoFinish, [33](#)
- scaleval< unsigned long long >
  - PhotoFinish, [34](#)
- scaleval< unsigned short int >
  - PhotoFinish, [34](#)
- set\_16bit
  - CMS::Format, [112](#)
- set\_32bit
  - CMS::Format, [112](#)
- set\_8bit
  - CMS::Format, [112](#)
- set\_alphaq
  - PhotoFinish::D\_JXR, [52](#)
- set\_artist
  - PhotoFinish::D\_TIFF, [70](#)
- set\_channel\_type
  - CMS::Format, [112–114](#)
- set\_chocolate
  - CMS::Format, [114](#)
- set\_colour\_model
  - CMS::Format, [114](#)
- set\_compression
  - PhotoFinish::D\_TIFF, [70](#)
- set\_copyright
  - CMS::Profile, [171](#)
  - PhotoFinish::D\_TIFF, [70](#)
- set\_defined
  - PhotoFinish::Role\_Definable, [175](#)
  - PhotoFinish::definable, [77](#)
- set\_depth
  - PhotoFinish::Destination, [85](#)
- set\_description
  - CMS::Profile, [171](#), [172](#)
- set\_double
  - CMS::Format, [114](#)
- set\_endianswap
  - CMS::Format, [114](#)
- set\_extra\_channels
  - CMS::Format, [115](#)
- set\_float
  - CMS::Format, [115](#)
- set\_half
  - CMS::Format, [115](#)
- set\_irreversible
  - PhotoFinish::D\_JP2, [44](#)
- set\_jp2
  - PhotoFinish::Destination, [85](#)
- set\_jpeg
  - PhotoFinish::Destination, [85](#)
- set\_jxr
  - PhotoFinish::Destination, [85](#)
- set\_lossless
  - PhotoFinish::D\_WebP, [72](#)
- set\_lossy
  - PhotoFinish::D\_WebP, [73](#)
- set\_manufacturer
  - CMS::Profile, [172](#)
- set\_method

- PhotoFinish::D\_WebP, 73
- set\_model
  - CMS::Profile, 172, 173
- set\_numresolutions
  - PhotoFinish::D\_JP2, 44
- set\_overlap
  - PhotoFinish::D\_JXR, 52
- set\_packed
  - CMS::Format, 115
- set\_planar
  - CMS::Format, 115
- set\_png
  - PhotoFinish::Destination, 86
- set\_premult\_alpha
  - CMS::Format, 116
- set\_preset
  - PhotoFinish::D\_WebP, 73
- set\_profile
  - PhotoFinish::Destination, 86
  - PhotoFinish::Image, 130
- set\_prog\_order
  - PhotoFinish::D\_JP2, 45
- set\_progressive
  - PhotoFinish::D\_JPEG, 49
  - PhotoFinish::D\_JXR, 52
- set\_qualities
  - PhotoFinish::D\_JP2, 45
- set\_quality
  - PhotoFinish::D\_JP2, 45
  - PhotoFinish::D\_JPEG, 49
  - PhotoFinish::D\_JXR, 52
  - PhotoFinish::D\_WebP, 73
- set\_rate
  - PhotoFinish::D\_JP2, 45
- set\_rates
  - PhotoFinish::D\_JP2, 45
- set\_resolution
  - PhotoFinish::Image, 130
- set\_resolution\_from\_size
  - PhotoFinish::Image, 130
- set\_reversible
  - PhotoFinish::D\_JP2, 46
- set\_sample
  - PhotoFinish::D\_JPEG, 49
- set\_sequential
  - PhotoFinish::D\_JXR, 52
- set\_subsampling
  - PhotoFinish::D\_JXR, 52
- set\_swap
  - CMS::Format, 116
- set\_swapfirst
  - CMS::Format, 116
- set\_tiff
  - PhotoFinish::Destination, 86
- set\_tile\_size
  - PhotoFinish::D\_JP2, 46
- set\_tilesizes
  - PhotoFinish::D\_JXR, 53
- set\_vanilla
  - CMS::Format, 116
- set\_webp
  - PhotoFinish::Destination, 86
- set\_xres
  - PhotoFinish::Image, 130
- set\_yres
  - PhotoFinish::Image, 131
- sharpen
  - PhotoFinish::Destination, 87
- sigma
  - PhotoFinish::D\_sharpen, 62
- size
  - PhotoFinish::D\_target, 64
  - PhotoFinish::Destination, 87
- solve
  - PhotoFinish::CropSolver, 41
- sqr
  - CropSolution.cc, 208
  - Kernel1Dvar.cc, 225
  - Kernel2D.cc, 226
- start
  - PhotoFinish::Timer, 184
- stop
  - PhotoFinish::Timer, 184
- StrPair
  - Tags.hh, 234
- stringlist
  - PhotoFinish, 22
- subsampling
  - PhotoFinish::D\_JXR, 53
- subst\_table
  - PhotoFinish, 22
- support
  - PhotoFinish::D\_resize, 60
- TIFFcheck
  - TIFFreader.cc, 236
  - TIFFwriter.cc, 237
- TIFFreader.cc, 236
  - TIFFcheck, 236
- TIFFwriter.cc, 236
  - TIFFcheck, 237
- Tags
  - PhotoFinish::Tags, 180
- Tags.cc, 232
- Tags.hh, 233
  - StrPair, 234
- Tags\_EXIF\_subst.cc, 234
  - Key, 234
- Tags\_IPTC\_subst.cc, 235
- Tags\_XMP\_subst.cc, 235
- targets
  - PhotoFinish::Destination, 87
- thumbnail
  - PhotoFinish::Destination, 87
- tiff
  - PhotoFinish::Destination, 87
- tile\_size



- PhotoFinish::D\_JP2, [46](#)
- tilesize
  - PhotoFinish::D\_JXR, [53](#)
- Timer
  - PhotoFinish::Timer, [184](#)
- total\_channels
  - CMS::Format, [116](#)
- transfer\_alpha
  - PhotoFinish, [34](#)
- transfer\_alpha\_typed
  - PhotoFinish, [34](#)
- transfer\_alpha\_typed2
  - PhotoFinish, [34](#)
- Transform
  - CMS::Format, [117](#)
  - CMS::Transform, [186](#)
- transform\_buffer
  - CMS::Transform, [188](#)
- transform\_colour
  - PhotoFinish::Image, [131](#)
- transform\_colour\_inplace
  - PhotoFinish::Image, [131](#)
- try\_load
  - PhotoFinish::Tags, [182](#)
- un\_alpha\_mult
  - PhotoFinish::Image, [132](#)
- undefine
  - PhotoFinish::Role\_Definable, [176](#)
  - PhotoFinish::definable, [77](#)
- Unimplemented
  - PhotoFinish::Unimplemented, [189](#)
- Uninitialised
  - PhotoFinish::Uninitialised, [191](#)
- UnknownFileType
  - PhotoFinish::UnknownFileType, [193](#), [194](#)
- unset\_endianswap
  - CMS::Format, [117](#)
- unset\_premult\_alpha
  - CMS::Format, [117](#)
- unset\_swap
  - CMS::Format, [117](#)
- unset\_swapfirst
  - CMS::Format, [117](#)
- variables
  - PhotoFinish::Tags, [182](#)
- warning\_callback
  - PhotoFinish, [35](#)
- waste
  - PhotoFinish::Frame, [121](#)
- WebP\_ostream.cc, [237](#)
  - min, [237](#)
- WebP\_ostream.hh, [238](#)
- WebP\_presets
  - PhotoFinish, [38](#)
- WebPError
  - PhotoFinish::WebPError, [198](#)
- WebPreader.cc, [238](#)
- WebPwriter.cc, [239](#)
- webp
  - PhotoFinish::Destination, [87](#)
- webp\_stream\_writer
  - PhotoFinish::webp\_stream\_writer, [195](#)
- webp\_stream\_writer\_func
  - PhotoFinish, [35](#)
- what
  - PhotoFinish::DestinationError, [89](#)
  - PhotoFinish::ErrorMsg, [96](#)
  - PhotoFinish::FileContentError, [98](#)
  - PhotoFinish::FileError, [100](#)
  - PhotoFinish::FileOpenError, [102](#)
  - PhotoFinish::LibraryError, [158](#)
  - PhotoFinish::MemAllocError, [159](#)
  - PhotoFinish::NoResults, [161](#)
  - PhotoFinish::NoTargets, [162](#)
  - PhotoFinish::Unimplemented, [190](#)
  - PhotoFinish::Uninitialised, [192](#)
  - PhotoFinish::UnknownFileType, [194](#)
  - PhotoFinish::WebPError, [199](#)
  - PhotoFinish::cmsTypeError, [40](#)
- width
  - PhotoFinish::D\_target, [64](#)
  - PhotoFinish::Image, [132](#)
- write
  - PhotoFinish::ImageWriter, [141](#)
  - PhotoFinish::SOLwriter, [178](#)
  - PhotoFinish::webp\_stream\_writer, [197](#)
- write\_be
  - PhotoFinish, [35](#)
- write\_chunk
  - PhotoFinish::webp\_stream\_writer, [197](#)
- write\_packed
  - PhotoFinish, [35](#)
- write\_planar
  - PhotoFinish, [36](#)
- XMP\_key\_subst
  - PhotoFinish, [38](#)
- XMPTags
  - PhotoFinish::Image, [132](#)
  - PhotoFinish::Tags, [183](#)
- xmp\_key\_read
  - PhotoFinish, [36](#)
- xres
  - PhotoFinish::Image, [132](#)
- yres
  - PhotoFinish::Image, [133](#)