NAME

intro – introduction to the XIL library

DESCRIPTION

These XIL reference manual pages describe the syntax for using the functions contained in the XIL Imaging Library. The library follows a standard "operator" imaging model. References to images (image handles) are passed to operators, which act on the image data. Each operator allows you to specify one or more source images and a single destination image, along with the parameters necessary to perform the operation. General information on how the library handles certain concepts is provided as follows:

Topic	*Reference Manual Page*
Images	**xil_create**(3)
Regions of interest	**xil_roi_create**(3)
Color spaces	**xil_set_colorspace**(3)
Origins	**xil_get_origin**(3)
Kernels	**xil_kernel_create**(3)
Overview on compressors	**xil_compress**(3)
Specific information on XIL compressors	Consult the man page for the specific compressor.

ERRORS

Error handling is asynchronous in the XIL library. Because operations are deferred to optimize groups of operations, errors may not be reported after the function call that produces the error is made in the application program. Instead, the error is reported when the set of operations that contains the error is executed.

The XIL library uses the following categories of errors:

User (or usage) errors
CIS data errors
Resource errors
Configuration errors
System errors
Arithmetic errors
Other errors

User Errors

Errors in this category are generated when a user passes invalid parameters to XIL functions or uses the library incorrectly in some other way.

CIS Data Errors

These errors occur when a bitstream does not conform to the specification of the specified compression type.

Resource Errors

The primary finite resource that the XIL library depends on is memory. If the XIL library runs out of memory, an error message to that effect will be generated. Then, depending on what the library was doing when the request for more memory occurred, a number of secondary error messages may also be generated, indicating the failure of the library to create objects, perform operations, or complete various other tasks. Some of these errors may be System errors (see below).

Configuration Errors

Errors in this category can occur if the XIL library is improperly installed, or a necessary environment variable is not set.

System Errors

XIL performs a number of internal checks on its operation. A failure of one of these checks is called an internal error. Internal errors should not occur. If such errors do occur (in the absence of an out-of-memory error), contact customer support and give as much information as possible about the error and the situation that caused it.

Arithmetic Errors

These errors occur when XIL detects an arithmetic erro in a program (for example, dividing by zero).

For More Information

Appendix B of the *XIL Programmer's Guide* provides a list of error messages by number. It also lists which XIL functions may generate a given error message. This section lists all the functions in the XIL Imaging Library. If the function does not appear on a man page bearing its name (in other words, it is grouped with other functions on a man page bearing the name of one of those other functions), then the function whose page it appears on is printed in parentheses beneath it.

Name *(Appears on Page)*	*Description*
Cell(3)	Cell compressor/decompressor for compressed image sequences (CISs)
CellB(3)	XIL driver for CellB video compression/decompression
faxG3(3)	CCITT Group 3 compressor for CISs
faxG4 (3) (**faxG3**(3))	CCITT Group 4 compressor for CISs
H261(3)	H.261 decompressor for CISs
Jpeg(3)	JPEG compressor/decompressor for CISs
JpegLL(3)	JPEG Lossless compressor/decompressor for CISs
Mpeg1(3)	MPEG decompressor for CISs

PhotoCD(3)	Reader for Kodak Photo CD(tm) format
xgl_to_xil (3) (**xil_to_xgl**(3))	converts XGL memory/display rasters to XIL memory/display images
xil_absolute(3)	finds the absolute value of pixels of an image
xil_add(3)	adds two images
xil_add_const (3) (**xil_add**(3))	adds a constant to each band of an image
xil_affine(3)	affine-transforms an image
xil_and(3)	bitwise logical AND operation
xil_and_const (3) (**xil_and**(3))	bitwise logical AND operation with constants
xil_band_combine(3)	interband linear combination operation
xil_black_generation(3)	adjusts amount of black in a CMYK image
xil_blend(3)	blends two images according to an alpha image
xil_call_next_error_handler (3) (**xil_install_error_handler**(3))	allows error handler further down the chain to handle the error
xil_cast(3)	casts an image from one data type into another
xil_choose_colormap(3)	chooses a reasonable colormap
xil_cis_attempt_recovery(3)	attempts recovery after an error occurs in a CIS
xil_cis_create(3)	creates a new CIS

xil_cis_destroy(3)	destroys a CIS
xil_cis_flush(3)	completes pending operations for a CIS
xil_cis_get_attribute(3)	gets a compressor attribute
xil_cis_get_autorecover(3)	indicates whether a decompressor will recover automatically from a recoverable datastream error
xil_cis_get_bits_ptr(3)	gets a pointer to compressed data
xil_cis_get_by_name(3)	returns a handle to the CIS object with the specified name
xil_cis_get_compression_type(3)	returns the name of the type of the compressor
xil_cis_get_compressor(3)	returns the name of a specific compressor
xil_cis_get_input_type(3)	returns the type of image that a CIS will accept for compression
xil_cis_get_keep_frames (3) (**xil_cis_get_max_frames**(3))	gets maximum number of previously decompressed frames in buffer
xil_cis_get_max_frames(3)	gets maximum number of compressed frames in buffer
xil_cis_get_name (3) (**xil_cis_get_by_name**(3))	returns a copy of the specified CIS object's name
xil_cis_get_output_type(3)	returns the XilImageType produced by a compressor
xil_cis_get_random_access(3)	shows whether a compressor supports random accessing of a CIS
xil_cis_get_read_frame (3) (**xil_cis_get_start_frame**(3))	returns index to the current frame

xil_cis_get_read_invalid(3)	determines whether a CIS is able to be decompressed
xil_cis_get_start_frame(3)	returns index to the first compressed image in the CIS
xil_cis_get_write_frame (3) (**xil_cis_get_start_frame**(3))	returns index to the last frame +1 of the CIS
xil_cis_get_write_invalid(3)	determines whether a CIS is able to continue to be compressed
xil_cis_has_data(3)	returns the number of bytes from the current frame to the end of the CIS
xil_cis_has_frame (3) (**xil_cis_has_data**(3))	returns TRUE if a complete frame exists at the read frame position
xil_cis_number_of_frames (3) (**xil_cis_has_data**(3))	determines number of complete unread frames of compressed data in the CIS
xil_cis_put_bits(3)	puts compressed data into a CIS
xil_cis_put_bits_ptr (3) (**xil_cis_put_bits**(3))	supplies a pointer to compressed data to a CIS
xil_cis_reset(3)	clears data in a CIS
xil_cis_seek(3)	finds a given frame of compressed data in a CIS
xil_cis_set_attribute (3) (**xil_cis_get_attribute**(3))	sets a compressor attribute
xil_cis_set_autorecover (3) (**xil_cis_get_autorecover**(3))	sets permission to attempt recovery if autorecoverable bitstream errors occur
xil_cis_set_keep_frames (3) (**xil_cis_get_max_frames**(3))	sets the number of frames prior to the current read frame to be kept in the buffer

xil_cis_set_max_frames (3) (**xil_cis_get_max_frames**(3))	sets the maximum number of frames or images in the buffer
xil_cis_set_name (3) (**xil_cis_get_by_name**(3))	sets the name of the specified CIS object to the one provided
xil_cis_sync(3)	forces any outstanding call to **xil_compress**(3) to complete when it would otherwise have been deferred
xil_close (3) (**xil_open**(3))	ends an XIL session
xil_color_convert(3)	converts an image from one color space to another
xil_colorcube_create(3)	creates a lookup table that represents a colorcube
xil_colorspace_get_by_name(3)	gets a color space object by its name
xil_compress(3)	compresses an image into a CIS
xil_convolve(3)	convolves an image with a specified kernel
xil_copy(3)	copies an image
xil_copy_pattern(3)	replicates the source image into the destination image
xil_copy_with_planemask(3)	use a plane mask to copy a source image into a destination image
xil_create(3)	creates an image
xil_create_child(3)	creates a child image
xil_create_copy(3)	creates a new image with a copy of the source's data
xil_create_from_device (3) (**xil_create_from_window**(3))	creates an image associated with the specified device

xil_create_from_type(3)	creates an image from an XilImageType object
xil_create_from_window(3)	creates an image associated with the specified X window
xil_decompress(3)	decompresses a CIS
xil_default_error_handler (3) (**xil_install_error_handler**(3))	prints error messages to the standard error output
xil_destroy(3)	destroys an image
xil_device_create(3)	creates a device object
xil_device_set_value(3)	stores device-initialization values in a device object
xil_device_destroy (3) (**xil_device_create**(3))	destroys a device object
xil_dilate (3) (**xil_erode**(3))	dilates an image
xil_dithermask_create(3)	creates a dither mask
xil_dithermask_create_copy (3) (**xil_dithermask_create**(3))	creates and returns a copy of the specified dither mask
xil_dithermask_destroy (3) (**xil_dithermask_create**(3))	destroys the specified dither mask
xil_dithermask_get_by_name(3)	returns a handle to the dither mask with the specified name
xil_dithermask_get_height(3)	gets the height of the specified dither mask
xil_dithermask_get_name (3) (**xil_dithermask_get_by_name**(3))	returns a copy of the specified dither mask's name
xil_dithermask_get_nbands (3) (**xil_dithermask_get_height**(3))	gets the number of bands in the specified dither mask

xil_dithermask_get_width (3) (**xil_dithermask_get_height**(3))	gets the width of the specified dither mask
xil_dithermask_set_name (3) (**xil_dithermask_get_by_name**(3))	sets the name of the specified dither mask to the one provided
xil_divide(3)	divides one image by another
xil_divide_by_const (3) (**xil_divide**(3))	divides a constant into each band of an image
xil_divide_into_const (3) (**xil_divide**(3))	divides each band of an image into constants
xil_erode(3)	erodes an image
xil_edge_detection(3)	detects edges within an image
xil_error_diffusion(3)	converts an image into a single-band image with a lookup table by error-diffusion dithering
xil_error_get_category (3) (**xil_error_get_string**(3))	returns the general category of the error
xil_error_get_category_string (3) (**xil_error_get_string**(3))	returns a character string that identifies the error category
xil_error_get_id (3) (**xil_error_get_string**(3))	returns a character string that uniquely identifies the error
xil_error_get_location (3) (**xil_error_get_string**(3))	returns an encrypted error location code
xil_error_get_object (3) (**xil_error_get_string**(3))	returns the XIL object that an error occurred on
xil_error_get_primary (3) (**xil_error_get_string**(3))	returns TRUE if the currently reported error is the primary cause of the error
xil_error_get_string(3)	returns an error string in the currently configured language

xil_export(3)	exports an image from XIL to application space
xil_extrema(3)	finds minimum and maximum values of an image
xil_fill(3)	performs boundary fill from a specified start point in an image
xil_get_attribute(3)	gets the values of client attributes of images
xil_get_by_name(3)	returns a handle to the image with the specified name
xil_get_child_offsets(3)	gets the values of the offsets into a parent image
xil_get_datatype(3)	gets an image's data type
xil_get_device_attribute(3)	gets the values of attributes of device images
xil_get_exported (3) (**xil_export**(3))	gets the export status of an image
xil_get_height (3) (**xil_get_width**(3))	gets the height of an image
xil_get_imagetype(3)	gets the type of an image
xil_get_info(3)	gets information about the parameters of an image
xil_get_memory_storage(3)	gets an image's memory storage
xil_get_name (3) (**xil_get_by_name**(3))	returns a copy of the specified image's name
xil_get_nbands (3) (**xil_get_width**(3))	gets the number of bands in an image
xil_get_origin(3)	gets the coordinates of the origin of an image

xil_get_origin_x (3) (**xil_get_origin**(3))	gets the x coordinate of the origin of an image
xil_get_origin_y (3) (**xil_get_origin**(3))	gets the y coordinate of the origin of an image
xil_get_parent(3)	gets a parent image
xil_get_pixel (3) (**xil_set_pixel**(3))	gets the value of a single pixel in an image
xil_get_readable(3)	returns TRUE if an image can be used as a source
xil_get_roi(3)	gets the region of interest (ROI) attached to an image
xil_get_size (3) (**xil_get_width**(3))	gets the size of an image
xil_get_synchronize (3) (**xil_sync**(3))	returns status of synchronization of an image
xil_get_width(3)	gets the width of an image
xil_get_writable (3) (**xil_get_readable**(3))	returns TRUE if an image can be used as a destination
xil_histogram(3)	generates histogram data from an image
xil_histogram_create(3)	creates a histogram object
xil_histogram_destroy (3) (**xil_histogram_create**(3))	destroys a histogram object
xil_histogram_get_by_name(3)	returns a handle to the histogram object with the specified name
xil_histogram_get_info (3) (**xil_histogram_get_nbands**(3))	gets values of histogram attributes

xil_histogram_get_limits (3) (**xil_histogram_get_nbands**(3))	gets values of arrays that represent values of the first and last bin in each band
xil_histogram_get_name (3) (**xil_histogram_get_by_name**(3))	returns a copy of the specified histogram object's name
xil_histogram_get_nbands(3)	gets the number of bands represented by the histogram
xil_histogram_get_nbins (3) (**xil_histogram_get_nbands**(3))	gets the array of values representing the number of histogram bins for each band
xil_histogram_get_values (3) (**xil_histogram_get_nbands**(3))	gets the array of values for the data attribute
xil_histogram_set_name (3) (**xil_histogram_get_by_name**(3))	sets the name of the specified histogram object to the one provided
xil_imagetype_get_by_name(3)	returns a handle to the image type object with the specified name
xil_imagetype_get_datatype(3)	gets an image type object's data type
xil_imagetype_get_height (3) (**xil_imagetype_get_width**(3))	gets the height of an image type object
xil_imagetype_get_info(3)	gets information about the parameters of an image type object
xil_imagetype_get_name (3) (**xil_imagetype_get_by_name**(3))	returns a copy of the specified image type object's name
xil_imagetype_get_nbands (3) (**xil_imagetype_get_width**(3))	gets the number of bands of an image type object
xil_imagetype_get_size (3) (**xil_imagetype_get_width**(3))	gets the size of an image type object
xil_imagetype_get_width(3)	gets the width of an image type object
xil_imagetype_set_name (3) (**xil_imagetype_get_by_name**(3))	sets the name of the specified image type object to the one provided

xil_import (3) (**xil_export**(3))	imports an image from application space to XIL space
xil_install_error_handler(3)	installs a customized error handler
xil_interpolation_table_create(3)	creates an interpolation table object
xil_interpolation_table_destroy (3) (**xil_interpolation_table_create**(3))	destroys an interpolation table object
xil_interpolation_table_get_data(3)	gets the data of an interpolation table object
xil_interpolation_table_get_kernel_size(3)	gets the kernel size of the subsample kernels in an interpolation table object
xil_interpolation_table_get_subsamples(3)	gets the number of subsamples in an interpolation table object
xil_kernel_create(3)	creates a kernel
xil_kernel_create_copy (3) (**xil_kernel_create**(3))	creates and returns a copy of the specified kernel
xil_kernel_destroy (3) (**xil_kernel_create**(3))	destroys the specified kernel
xil_kernel_get_by_name(3)	returns a handle to the kernel object with the specified name
xil_kernel_get_height(3)	gets the height of a kernel
xil_kernel_get_key_x (3) (**xil_kernel_get_height**(3))	gets the x coordinate of the key value of a kernel
xil_kernel_get_key_y (3) (**xil_kernel_get_height**(3))	gets the y coordinate of the key value of a kernel
xil_kernel_get_name (3) (**xil_kernel_get_by_name**(3))	returns a copy of the specified kernel object's name
xil_kernel_get_width (3) (**xil_kernel_get_height**(3))	gets the width of a kernel

xil_kernel_set_name (3) (**xil_kernel_get_by_name**(3))	sets the name of the specified kernel object to the one provided
xil_lookup(3)	passes an image through a lookup table
xil_lookup_convert(3)	calculates a lookup table that converts between source and destination lookup tables
xil_lookup_create(3)	creates a single lookup table
xil_lookup_create_combined(3)	creates a combined lookup table
xil_lookup_create_copy (3) (**xil_lookup_create**(3))	creates and returns a copy of the specified lookup table
xil_lookup_destroy (3) (**xil_lookup_create**(3))	destroys a lookup table
xil_lookup_get_band_lookup(3)	gets a single lookup table out of a combined lookup
xil_lookup_get_by_name(3)	returns a handle to the lookup table with the specified name
xil_lookup_get_colorcube (3) (**xil_colorcube_create**(3))	returns TRUE if a lookup table is formatted as a colorcube
xil_lookup_get_colorcube_info (3) (**xil_colorcube_create**(3))	returns formatting information about a lookup table used as a colorcube
xil_lookup_get_input_datatype(3)	gets the data type of the input to a lookup table
xil_lookup_get_name (3) (**xil_lookup_get_by_name**(3))	returns a copy of the specified lookup table's name
xil_lookup_get_num_entries (3) (**xil_lookup_get_input_datatype**(3))	gets the number of entries in a lookup table
xil_lookup_get_offset (3) (**xil_lookup_get_input_datatype**(3))	gets the offset value of a lookup table

xil_lookup_get_input_nbands (3) (**xil_lookup_get_output_nbands**(3))	gets the number of bands in the input from a lookup table
xil_lookup_get_output_datatype (3) (**xil_lookup_get_input_datatype**(3))	gets of data type of the output from a lookup table
xil_lookup_get_output_nbands (3) (**xil_lookup_get_input_datatype**(3))	gets the number of bands in the output from a lookup table
xil_lookup_get_values (3) (**xil_lookup_set_values**(3))	gets the values in a lookup table
xil_lookup_get_version(3)	gets the unique version number of a lookup table
xil_lookup_set_name (3) (**xil_lookup_get_by_name**(3))	sets the name of the specified lookup table to the one provided
xil_lookup_set_offset (3) (**xil_lookup_get_input_datatype**(3))	sets the offset value of a lookup table
xil_lookup_set_values(3)	sets the values in a lookup table
xil_max(3)	finds the larger of pixels in two images
xil_min(3)	finds the lesser of pixels in two images
xil_multiply(3)	multiplies two images
xil_multiply_const (3) (**xil_multiply**(3))	multiplies each band of an image by a floating point constant
xil_nearest_color(3)	converts an image into a single-band image by mapping pixels to the nearest entries in a lookup table
xil_not(3)	bitwise logical NOT operation
xil_object_get_error_string (3) (**xil_error_get_string**(3))	creates a string with additional information about the object involved in the error

xil_object_get_type (3) (**xil_error_get_string**(3))	returns the XilObjectType of an object
xil_open(3)	opens the XIL library for use
xil_or(3)	bitwise logical OR operation
xil_or_const (3) (**xil_or**(3))	bitwise logical OR operation with constants
xil_ordered_dither(3)	uses ordered dithering to convert an image into a single-band image with a lookup table
xil_paint(3)	blends portions of an image with a single color using a 2-D brush
xil_remove_error_handler (3) (**xil_install_error_handler**(3))	removes an error function from the error handler chain
xil_rescale(3)	rescales an image
xil_roi_add_image(3)	adds a binary image to an ROI
xil_roi_add_rect(3)	adds a rectangle to an ROI
xil_roi_add_region(3)	adds an X region to an ROI
xil_roi_create(3)	creates an ROI
xil_roi_create_copy (3) (**xil_roi_create**(3))	creates and returns a copy of an ROI
xil_roi_destroy (3) (**xil_roi_create**(3))	destroys an ROI
xil_roi_get_as_image(3)	gets an image version of an ROI
xil_roi_get_as_region(3)	returns a handle to an X region
xil_roi_get_by_name(3)	returns a handle to the ROI with the specified name

xil_roi_get_name (3) (**xil_roi_get_by_name**(3))	returns a copy of the specified ROI's name
xil_roi_intersect(3)	finds the intersection of two ROIs
xil_roi_set_name (3) (**xil_roi_get_by_name**(3))	sets the name of the specified ROI to the one provided
xil_roi_subtract_rect(3)	subtracts a rectangle from an ROI
xil_roi_translate(3)	translates an ROI
xil_roi_unite(3)	finds the union of two ROIs
xil_rotate(3)	rotates an image
xil_scale(3)	scales an image
xil_sel_create(3)	creates a structuring element (SEL)
xil_sel_create_copy (3) (**xil_sel_create**(3))	creates and returns a copy of a SEL
xil_sel_destroy (3) (**xil_sel_create**(3))	destroys a SEL
xil_sel_get_by_name(3)	returns a handle to the SEL with the specified name
xil_sel_get_height(3)	gets the height of a SEL
xil_sel_get_key_x (3) (**xil_sel_get_height**(3))	gets the x coordinate of the key value of a SEL
xil_sel_get_key_y (3) (**xil_sel_get_height**(3))	gets the y coordinate of the key value of a SEL
xil_sel_get_name (3) (**xil_sel_get_by_name**(3))	returns a copy of the specified SEL's name
xil_sel_get_width (3) (**xil_sel_get_height**(3))	gets the width of a SEL

xil_sel_set_name (3) (**xil_sel_get_by_name**(3))	sets the name of the specified SEL to the one provided
xil_set_attribute (3) (**xil_get_attribute**(3))	sets the values of client attributes of images
xil_set_colorspace(3)	sets an image's color space
xil_set_device_attribute (3) (**xil_get_device_attribute**(3))	sets the values of attributes of device images
xil_set_memory_storage (3) (**xil_get_memory_storage**(3))	sets an exported image's memory storage
xil_set_name (3) (**xil_get_by_name**(3))	sets the name of the specified image to the one provided
xil_set_origin (3) (**xil_get_origin**(3))	sets the coordinates of the origin of an image
xil_set_pixel(3)	sets the value of a single pixel in an image
xil_set_roi (3) (**xil_get_roi**(3))	sets an image's ROI
xil_set_synchronize (3) (**xil_sync**(3))	sets synchronization of an image
xil_set_value(3)	sets pixels of an image to constant values
xil_soft_fill(3)	performs a soft fill from a specified starting point in an image
xil_squeeze_range(3)	produces a lookup table that will map an image into contiguous entries
xil_state_get_interpolation_tables(3)	gets interpolation tables from the XilSystemState object
xil_state_set_interpolation_tables (3) (**xil_state_get_interpolation_tables**(3))	sets interpolation tables on the XilSystemState object

xil_state_get_show_action(3)	gets the current value of the SHOW_ACTION attribute of a system-state object
xil_state_get_synchronize (3) (**xil_sync**(3))	returns synchronization status of an XIL State
xil_state_set_show_action (3) (**xil_state_get_show_action**(3))	sets the current value of the SHOW_ACTION attribute of a system-state object
xil_state_set_synchronize (3) (**xil_sync**(3))	sets synchronization status for an XIL State
xil_subsample_adaptive(3)	adaptively subsamples an image
xil_subsample_binary_to_gray(3)	subsamples a binary image and produces a grayscale image
xil_subtract(3)	subtracts one image from another
xil_subtract_const (3) (**xil_subtract**(3))	subtracts a constant from each band of an image
xil_subtract_from_const (3) (**xil_subtract**(3))	subtracts each band of an image from a constant
xil_sync(3)	forces computation of the value of an image when it would have otherwise been deferred
xil_tablewarp(3)	warps an image in both the horizontal and vertical directions
xil_tablewarp_horizontal (3) (**xil_tablewarp**(3))	warps an image in the horizontal direction
xil_tablewarp_vertical (3) (**xil_tablewarp**(3))	warps an image in the vertical direction
xil_threshold(3)	sets value of image pixel bands within a specified range

xil_to_xgl(3)	converts XIL memory/display images to XGL memory/display rasters
xil_toss(3)	throws away the contents of an image without destroying it
xil_translate(3)	translates an image
xil_transpose(3)	rotates or transposes an image
xil_xor(3)	bitwise logical XOR operation
xil_xor_const (3) (**xil_xor**(3))	bitwise logical XOR operation with constants