

# Cyclops 1.3 User's Guide

©2001 Code Artistry LLC. All rights reserved.

Updates ©2003-2008 Cycling '74

cyclops-info@ericsinger.com

Cyclops is a Max object which receives and analyzes video input. It receives input from a QuickTime video input source (such as a video capture card, USB or FireWire video input), analyzes the video frames and outputs messages based on the images.

To use Cyclops, you specify areas of the image to analyze (called 'zones'), and what type of analysis to perform - greyscale, threshold, difference (motion) or color. Cyclops can then capture and analyze frames of incoming video and output messages based on the results.

If you are familiar with VideoIn, Cyclops's predecessor, you will notice that Cyclops has been completely redesigned. Cyclops adds color support, a graphical user interface and editing features, a better zone numbering system and saving of video and object settings. It works properly with most QuickTime input hardware.

Cyclops is available as a full-featured time-limited demo. After purchase and authorization, the demo version converts to the regular (unlimited use) version. Though Cyclops works with most systems and QuickTime hardware, you should test it with your system using the demo version before purchasing.

jit.cyclops is a version of Cyclops modified for Jitter interoperability. It shares most of Cyclops's capabilities, but analyzes Jitter matrices.

## **Changes in version 1.3.2**

- Windows version number bump (to 1.3.2 from 1.2.2).
- Max versions < 4.6 no longer supported.
- Runtime usage of Cyclops was improperly limited to 2 zone operation. Fixed.

## **Changes in version 1.2.3/1.3.1**

- Attempting to open the Settings Dialog for an inactive Cyclops object no longer causes a crash.
- Opening 2 instances of Cyclops reports a warning in the Max Window.

## **Changes in version 1.3 (Mac OSX Only)**

- Universal Binary version.

## **Changes in version 1.2.2**

- Fixed problems with Zone Dialog.

## **Changes in version 1.2.1**

- Fixed Runtime behavior - in version 1.2, the object was not functioning properly under the Runtime.
- Fixed a memory corruption bug with the Delete Zones dialog.

**NOTE:** Due to changes in the Copy Protection in version 1.2.1, patches saved with previous versions of Cyclops or jit.cyclops *will not* work in the Runtime with version 1.2.1 (or higher) of the objects. The patch must be re-opened using the full version of MaxMSP and re-saved.

### **Changes in version 1.2**

- The software now functions in the Runtime without an authorization, with the restriction that the creation of new zones is not permitted.
- Compiled for Windows XP systems running Max 4.5.x (and optionally Jitter 1.5.x).
- Eliminated system dialog boxes in favor of inspector-like patches
- Removed support for Classic MacOS

**NOTE:** on Windows systems, your video input hardware **must** either have a QuickTime driver, or function with a Video Digitizer component (VDIG) providing a conversion layer between DirectX and QuickTime. At the time of this writing, we are aware of two such VDIGs: WinVDIG ([www.vdig.com](http://www.vdig.com)) and Abstract Plane ([www.abstractplane.com.au](http://www.abstractplane.com.au))

### **Changes in version 1.1**

- Code review and cleanup.
- New messages: 'videopos' and 'palpos' can be used to position windows using messages.
- Bug fix: Some input devices would have their image cropped. Fixed.
- Enhancement: OSX support.
- Enhancement: Jitter support. See the jit.cyclops chapter at the end of this manual for more information.
- Changes to the copy protection.

### **Installation**

Please refer to the Cyclops Read Me file that came with Cyclops for installation and authorization instructions.

## Windows

Cyclops has two windows: Video and Palette. The Video window displays unprocessed or processed video frames. The 'display' message sets the display mode ('off', 'grey', 'diff', 'color' or 'input'). The Video window is also used for creating and editing analysis zones and for sampling colors.

You can set the size of the Video window with the 'videosize' message or by using the mouse. Holding the shift key while resizing with the mouse constrains the window to "standard" frame sizes: 80 x 60, 160 x 120, 320 x 240 or 640 x 480.

You can set the position of the Video or Palette windows with the 'videopos' and 'palpos' messages, respectively.

The Palette window displays target colors for use in color analysis modes. Double-clicking on a color opens an editing dialog which you can use to set the color range which it represents. Colors can also be set by capturing them from the Video window and transferring them to the Palette ("sampling").

Double clicking on the Cyclops object box opens the Video and Palette windows if they are closed and brings them to the front.

## Blocks and Zones

Cyclops divides the video image into a grid of blocks. Blocks are equally-sized rectangles created by dividing the image into the grid. The dimensions of the grid are set with the 'grid <x> <y>' message, where x and y are the number of blocks across and down.

Zones are locations in the video frame. Zones are created by command-clicking in the Video window, which opens the zone editing dialog and places a dot for the zone location. The zone location determines which block will be analyzed for that zone. The zone settings determine what type of analysis will be performed.

When the 'grid' dimensions are changed, Cyclops determines which block each zone lies in. If you create a zone, then change the 'grid' dimensions, you can see how the block containing the zone will change.

Zone locations are not affected by the size of the Video window. For example, if a zone is 25% from the left edge and 25% from the top edge of the window, it will remain in this relative location (and thus in the same block) if the Video window is resized.

To process video, Cyclops averages the pixels within each block, yielding an 8-bit grey value and/or 24-bit color value for each block. A zone's output is determined by analyzing the averaged pixel value of its corresponding block to generate an output value or values.

A zone output message consists of the zone number followed by the data value(s). All enabled zones are analyzed once per input frame. By default, zone messages are output once per frame only when their values change. You can change this behavior using the 'changemode' and 'bangmode' messages (see "Output modes"). Messages are output in order of increasing zone number.

It is possible (and perfectly acceptable) to have a block which contains more than one zone. You would do this if you wanted two or more types of analysis reported for the same block. (Note that you can not have more than one zone with the same zone number - that is, zone numbers must be unique.)

## Creating and editing zones

Zones are created and edited by using the mouse in the Video window and with the zone editing dialog box. Zones can also be created and edited with messages.

To create a new zone, command-click on the Video window at the point where you want to place the zone. A dialog box opens and presents options for setting zone parameters. To edit an existing zone, double-click on it to reopen the dialog box.

Zones can be moved and deleted using standard mouse methods. Click on a zone to select it. Shift-click to select multiple zones, or click and drag to use a marquee rectangle to select. Click and drag a zone to move it, along with any other selected zones. Hit the delete key (with the Video window active) to delete selected zones. The 'clear' message deletes all zones.

Zones are drawn as a dot with a zone number adjacent. The zone number is shown to the right of the zone dot (unless it is too close to the righthand edge of the window, in which case it is drawn to the left). Selected zones are shown with a circle around them.

To set the maximum number of zones, use the 'maxzones' message. The default is 256 zones.

## Cyclops analysis process

As each video frame is captured, Cyclops performs the following analysis sequence. The captured frame is converted into one or two low resolution versions - one for color and one for greyscale analysis (as necessary; see below). This high-resolution (capture) to low-resolution (analysis) conversion has the effect of averaging blocks of grey or color information in the image. For each analysis zone, Cyclops looks at the block which contains the zone, performs the requested analysis and computes an output value (or values, in the case of 'color' mode).

After all zones are analyzed, Cyclops sends a 'sync 1' message out the right outlet. If 'bangmode' is off (the default state), zone messages are then sent out the left outlet in increasing zone number order. If 'changemode' is on (the default state), messages are output for zones with values that changed (since they were last output). If 'changemode' is off, messages are output for all zones. Cyclops sends a 'sync 0' message out the right outlet when it has finished outputting zone messages.

Finally, the analysis frame is copied to the output window in the manner specified by the 'display' message.

For all messages except for zones in 'color' mode, the zone message format is '<n> <d>', where n is the zone number and d is the data value. The interpretation of the data value is dependent on the zone mode. For zones in 'color' mode, output format is '<n> <r> <g> <b>', where r, g, b are red, green and blue values for the zone.

Greyscale and color resolution conversions are performed only if necessary. If *any* zone is set to a greyscale analysis mode ('grey', 'diff', 'thresh' or 'diffthresh'), or if the display mode is 'grey' or 'diff', Cyclops performs the greyscale conversion. If *any* zone is set to a color analysis mode ('color', 'match',

'closest', or 'closestmatch'), or if the display mode is 'color', Cyclops performs the color conversion. (A display mode of 'input' or 'off' does not affect this determination).

In other words, Cyclops performs only the conversions necessary under the conditions you have set. You can increase Cyclops's efficiency by causing it to perform only one conversion. Use either greyscale or color analysis modes, but not both, and set the display mode to correspond.

## Analysis modes

Cyclops can perform many types of analysis. Analysis type is set individually for each zone. The following is a description of the available analysis modes.

‘grey’ mode outputs a grey value from 0 to 255. 0 is black, 255 is white and values in between are shades of grey.

‘diff’ mode outputs the difference in grey values between the current and the previous frame. Changes from dark to light output positive values; changes from light to dark output negative values. For example, if a zone changes from black (0) to white (255), the output will be 255; if the change is reversed, the output will be -255.

‘thresh’ mode outputs a 1 or 0 when the grey value crosses a specified threshold. 1 is output when the threshold level is crossed in the positive (white) direction; 0, when crossed in the negative (black) direction.

‘diffthresh’ mode outputs a 1, 0 or -1 when the difference value crosses a specified threshold. 1 is output when the difference value crosses the threshold value in the positive (black to white) direction; -1 is output when the difference value crosses the *negative* of the threshold value in the negative (white to black) direction; 0 is output when the difference value crosses back between the +/- threshold value. For example, say the threshold is 10. If the difference goes from 0 to 11, a 1 is output. If it goes back down to 9, a 0 is output. If it now goes down to -11, a -1 is output. If it then increases to -9, a 0 is output.

‘color’ mode outputs three values, corresponding to red, green and blue levels. Each value is an 8-bit color level, representing the levels of red, green and blue (respectively) in the zone. ‘color’ is the only mode which outputs more than a one value per message.

The remaining three modes, ‘match’, ‘closest’ and ‘closestmatch’, use the Palette colors as target values to look for in a zone. Target colors are specified using Palette color numbers. When selecting one of these modes, you also specify a list of color numbers comprising the target colors. (When entering a color list in the zone editing dialog, separate color numbers with spaces.)

‘match’ mode compares the color value in the zone with the target colors. It outputs the color number for the first color in the list which is a match. A color value matches a target value if all three color components (red, green and blue) are within the specified +/- range of target’s color components.

For example, say a target color has an R/G/B value of 10/20/30, with +/- values of 1/2/3 (respectively). A captured color value of 9/21/32 would match this target, but 8/21/32 would not (the red value is outside the target’s red range).

If no value in the target list is a match for the captured color value, ‘match’ outputs -1.

‘closest’ mode outputs the number of the target color which is closest to the captured color. It determines the closest color by using a formula to check the distance between the captured color and each target color. It does not use the +/- values of the target, only the color value. For those interested in the arcane details,

the formula is  $\text{distance} = \sqrt{(\Delta R)^2 + (\Delta G)^2 + (\Delta B)^2}$ . 'closest' mode will *always* output one of the target colors in the color list.

'closestmatch' is a combination of 'closest' and 'match' mode. It finds all target colors which satisfy the 'match' criteria, and outputs the number of the closest among the matches. If no targets match, it outputs -1.

## **Output modes and 'sync' messages**

Two messages, 'bangmode' and 'changemode', affect how and when Cyclops outputs zone messages. With 'bangmode 0' (the default), Cyclops outputs zone messages once per captured video frame. With 'bangmode 1', Cyclops outputs zone messages only when it receives a bang. A bang at any time will output the zone messages for the most recently analyzed frame. By setting 'bangmode 1' and banging Cyclops with a metro, you can control the data output rate.

With 'changemode 1' (the default), Cyclops outputs zone messages only for zones with values that changed since last output. With 'changemode 0', Cyclops outputs messages for all zones every time it outputs.

Each time Cyclops is finished analyzing a frame, it outputs a 'sync 1' message. When Cyclops finishes outputting zone messages (after each analysis in 'bangmode 0', or after each bang in 'bangmode 1'), it outputs a 'sync 0' message. You can use the sync message to synchronize your Max processing to the moment when Cyclops analysis and data output have finished.

## **The Palette and palette colors**

The Palette holds up to 100 colors and is used by Cyclops's color matching modes ('match', 'closest' and 'closestmatch'). Palette colors are shown in the Palette window and are numbered consecutively from 0 to 99, going from left to right, top to bottom. 100 is probably more colors than you will ever need, but they are there if you do. Generally, you will only specify a handful of colors.

A color in the Palette is actually a color range. Each Palette color has three values comprising the color center (referred to simply as the "color") and three values comprising the range or allowable variation in color (referred to as the "variation").

The three values comprising the color are 8-bit red, green and blue (RGB) color components. These are the values you see in the Mac's RGB color picker. The color created by these values is displayed in the corresponding location in the Palette window. Colors which are unassigned (either never given values or deleted) display as a grey diagonal pattern.

The three values comprising the variation are the amount of allowable difference from each of the RGB components. These values are specified as +/- variation from the RGB components.

For example, a Palette color with RGB values (in percent) of 100/40/20 and +/- values of 5/10/15 would display as a shade of orange in the Palette. For color matching purposes, this color would allow for 5%, 10% and 15% variation in red, green and blue values (respectively). This would include all colors with a red value between 95% and 100%, green between 30% and 50%, and blue between 5% and 35%. Note that values pin between 0% and 100%, or 0 to 255. Thus in this example, maximum red is 100%, not 105%.

To view and edit color values, double-click on a Palette color. This opens a dialog box with edit boxes for 6 values (RGB color values and +/- amounts). The values can be viewed and edited as either absolute 8-bit values (0-255) or as percentages (0-100). Also, you can edit the color with the color picker by clicking the Pick button. To set a color back to unassigned, click the Delete button.

## **Color sampling**

In addition to editing Palette colors directly, you can sample colors from incoming video and assign them to the Palette. The sampling process works as follows. First, start video and enable sampling. Then, use the mouse in the Video window to acquire samples from the incoming video. Finally, assign the sampled data to a Palette color. This is explained in detail in the section "How to sample colors".

Taking a number of samples allows you to establish a color range. The color of an item as viewed by a video camera is affected by many factors, such as its location in the image, the lighting conditions and its angle relative to lights and camera. By moving an item around in the image and taking numerous samples, you can establish a range of color variation.

When you assign the sampled data to a Palette color, Cyclops performs calculations to determine a color center (average) and a range of variation. It attempts to factor in samples which are close to the average and disregard samples which are far away. In this way, it makes its best estimate of the color and variation you

are trying to capture.

Note that these calculations are only as good as the data you have sampled. If you take samples of areas with unrelated colors, the calculations will yield useless values.

If you take only a single sample of a color, it will be assigned to the Palette with a variation of 0 for each RGB component.

### **How to sample colors**

To enable sampling, start video and send the message 'sampling 1'. You will probably also want to set the display mode to 'color' or 'input'. In color mode, each block displays the color that will be sampled when you click in the block.

Clicking in a block in the Video window samples the block color from the processed video input. Clicking takes a single sample; shift-clicking and holding takes continuous samples. The number of samples acquired is displayed in the title bar.

After you have acquired one or more samples, click on a location in the Palette to assign the color. You can then double-click on the color, view the data and edit it if you choose.

To discard acquired samples and reset the sample count to 0, press the delete key with the Video window active. After assigning a color to the Palette, the sample count is automatically reset to 0. Remember to turn off sampling when you are finished, by sending the message 'sampling 0'.

## The 'capsize' message

'capsize' changes the resolution at which Cyclops captures video input frames (prior to processing them). 'capsize <x> <y>' sets the capture resolution to x by y pixels. The default capture resolution is 320 x 240.

You may be able to affect Cyclops's efficiency by changing the capture resolution. Generally, lowering the resolution will make Cyclops more efficient. However, some input devices can only capture video at certain resolutions. If the hardware you are using does not support the resolution you select, you will likely see only part of the video frame; part of the Video window will be blank or contain garbage.

Try changing 'capsize' to see if a lower resolution will work with your hardware. If it does, you will likely see decrease in latency and/or an increase in frame rate.

It is conceivable that some devices will be more efficient using their "native" capture resolution (for example, DV native resolution is 720 x 480) than a lower resolution. Since video input hardware and drivers vary, it is best to experiment.

## QuickTime processing and the 'idlerate' message

In order for Cyclops to acquire video frames, it must give processing time to QuickTime. Internally, this is accomplished by periodically calling a QuickTime "idle" routine. The rate at which this is done is set with the 'idlerate' message.

In normal operation, 'idlerate' is set to 0. This tells Cyclops to give time to QuickTime once each time through Max's internal event loop. Though this may not mean much to you, to Max, it means to call this routine at the optimum rate. Note that this rate is not necessarily constant.

If you send an 'idlerate <t>' message (where t is not 0), it will cause Cyclops to give time to QuickTime once every t milliseconds. Depending on the value of t, this could have the effect of calling the QuickTime routine more or less often than with 'idlerate 0'.

In most circumstances, you will not need or want to change 'idlerate'. However, if you have certain problems with normal operation, you can try to alleviate them by changing 'idlerate'.

If video processing seems to be slowing everything else down, such as mouse response or other Max operation, you can change 'idlerate' to give time to QuickTime less often. You might begin by trying a value of 500 ms, and adjusting this up or down to see the effect.

Note that 'idlerate' and video frame rate are not equivalent. It is possible in some circumstances that you can use 'idlerate' to control frame rate; however, if you require analysis data to be reported at a particular rate, you should store output data and clock it out at the desired rate.

Using 'idlerate' to attempt to speed up video processing *is not likely to help* and may in fact unduly slow down the rest of Max's operation. You are welcome to try this, though, as results will vary depending on which Mac and video hardware you are using. Note that you can easily crash Max by setting 'idlerate' too low.



## Improving performance

Video frame capture and processing can take a lot of CPU power. Here are some ways you can affect the efficiency of Cyclops's capturing and processing to attempt to improve performance (frame rate and latency).

- Close the Video window or set the display mode to 'off' when you are not using it.
- Reduce the size of the Video window.
- Reduce the capture resolution using 'capsize'. Note that some input devices can only capture properly at certain resolutions.
- Use either grey *or* color analysis and display modes, but not both.
- Turn off drawzones and drawgrid.
- If available, try using various input devices to see which one is most efficient. Efficiency and latency can vary a lot depending on hardware. For example, as of this writing, FireWire input is known to have latency problems. The Irez CapSure PCM-CIA card works very well and efficiently with Cyclops and is a recommended solution. Internal video capture boards and cards will probably tend to work better than USB, FireWire and serial devices, though this has not been tested.

## Settings saved with patch

When you save a patch containing a Cyclops object, the following information is saved with the patch: window dimensions and locations, zone settings, palette settings, video settings, and settings for 'grid', 'display', 'drawzones', 'drawgrid', 'zonecolor', 'gridcolor', 'bangmode', 'changemode' and 'maxzones'.

You can control the save feature using the 'savemode' message. With 'savemode 2' (the default), all settings are saved. With 'savemode 1', only video input settings are saved. With 'savemode 0', no settings are saved. You should send the 'savemode' message to Cyclops right after opening the patch, because the mode applies to changes made after the message is received. Loadbanging the savemode message is a good idea.

## Other issues and features

For display purposes, Cyclops assumes that you are using 24-bit color (the "millions of colors" mode on your display). If you are using a lower color depth, the colors you see may not correspond to their real values, and windows may draw with strange colors.

You will notice a flicker in the Video window when video is running and 'drawzones' and/or 'drawgrid' is enabled. For efficiency reasons, this flicker has purposely not been eliminated, as it would require extra drawing operations.

Cyclops's mouse, key and menu editing features are disabled when video is running. If you try to use these features while running, Cyclops will inform you by printing a message in the Max window.

Holding down the control key and the mouse button stops video. You can use this to stop video in a pinch (e.g. if the processor is bogged down and mousing becomes difficult).

You should avoid moving the Video window while video is running. Usually, it will stop video capture and analysis. You can generally restart capture by toggling capture off and on again (0, 1)

## Message descriptions

1	start video
0	stop video
bang	output zone data (use when 'bangmode 1' is set)
settings	open video settings dialog
listdevices	list available QuickTime input devices
device	report current device
device <n>	set input device to device n (where n is a number from 'listdevices')
inputnum <n>	set input on current device to input n (where n is from 1 to the number of inputs on the current device)
composite	set input to the composite input on the current device (or the first composite input if there is more than one)
svideo	set input to the S-video input on the current device (or the first S-video input if there is more than one)
rgbcomponent	set input to the RGB component input on the current device (or the first RGB component input if there is more than one)
grid <x> <y>	set the dimensions of the zone grid to x by y
videosize <x> <y>	set the size of the Video window to x by y pixels
videopos <h> <v>	move the Video window to screen coordinates [h, v]
palpos <h> <v>	move the Palette window to screen coordinates [h, v]
display <mode>	set the display mode for the Video window, where mode is one of the following: off: turn off display color: display color data grey: display grey data diff: display grey difference data; a difference value of 0 displays as medium grey, 255 as white and -255 as black input: display the input video signal
drawzones <b>	enable (b = 1) or disable (b = 0) drawing of zones
drawgrid <b>	enable (b = 1) or disable (b = 0) drawing of grid
zonecolor	open color picker to pick zone drawing color
gridcolor	open color picker to pick grid drawing color
videowin <b>	open (b = 1) or close (b = 0) Video window
palwin <b>	open (b = 1) or close (b = 0) Palette window
zone <n> <mode> (<d1> <d2> ...)	

edit settings for a zone, where n is the zone number, mode is the analysis mode, and d1, d2, ... are data values dependent on the mode; modes 'off', 'grey', 'diff' and 'color' have no data; for 'thresh' and 'diffthresh', d1 is the threshold (0-255); for modes 'match', 'closest' and 'closestmatch', d1, d2, ... are color numbers (0-99)

zonexy <n> <locx> <locy> <mode> (<d1> <d2> ...)

create a new zone or move and edit an existing zone; format is the same as the 'zone' message, with the addition of locx and locy, which are the zone's x and y location in the window as values from 0.0 to 1.0 (x, y = 0.0 is left, top, x,y = 1.0 is bottom, right); if <n> is an existing zone number, the zone is edited, else a new zone is created

alloff set all zones to 'off'

clear delete all zones

zoneinfo send info about all zones out the right outlet; the format is <n> <gridx> <gridy> <locx> <locy> <mode> (<d1> <d2> ...), where n is the zone number, gridx and gridy are the zone's x and y location in the grid, locx and locy are the zone's x and y location in the window as values from 0.0 to 1.0 (x, y = 0.0 is left, top, x,y = 1.0 is bottom, right), and mode, d1, d2, ... are the same as in the 'zone' command

palette <n> <r> <g> <b> <rvar> <gvar> <bvar>

set color and variation values for palette color n, where r/g/b are the color values (0-255) and rvar/gvar/bvar are the variation values

sampling <b> enable (b = 1) or disable (b = 0) color sampling

bangmode <b> output data each frame(b = 0) or only when a bang is received(b = 1)

changemode <b> output data for zones with changed values (b = 0) or for all zones each time(b = 1)

capsize <x> <y> set the capture resolution to x by y pixels

idlerate <t> allow QuickTime processing time every t milliseconds; t = 0 (default) gives processing time to QuickTime once per event loop

maxzones <n> set the maximum number of zones which can be created

savemode <n> n = 0 means don't save settings; n = 1 means save video input settings only; n = 2 means save all settings

## **jit.cyclops**

New in version 1.1 of Cyclops is compatibility with Jitter, with the **jit.cyclops** object. Using the `jit.cyclops` object requires Jitter, and may have additional system requirements, as specified in the Cyclops Read Me.

In general, the `jit.cyclops` object functions identically to the Cyclops object, with the following differences:

**Sequence grabber functions are disabled:** `jit.cyclops` uses Jitter for video input, and accepts a 1 or 4 plane char matrix in its inlet, along with any messages you'd send to the object. `jit.cyclops` processes the incoming matrix exactly as if it were video captured by the Cyclops object. To use `jit.cyclops` as though it were the standard Cyclops object, you could connect a `'jit.qt.grab'` object to the `jit.cyclops` object's inlet. To process a QuickTime movie, you could connect a `'jit.qt.movie'` object, etc.

**New outlet:** The object has one new outlet: the leftmost outlet is used to output the same visual data seen in the Video Window, without the gridlines or the zone markers, as a Jitter matrix.

**Idling:** The `jit.cyclops` object doesn't support `'idlerate'`, and simply functions at whatever rate it receives matrices.

**Capture Size:** The `jit.cyclops` object doesn't support `'capsize'`, using the standard Jitter `'dim'` message to set the resolution of the internal Jitter matrix.

**Message changes:** The following messages have no meaning, and have been disabled, in the `jit.cyclops` object:

settings  
listdevices  
device  
inputnum  
composite  
svideo  
rgbcomponent  
idlerate  
capsize

The following messages have a different meaning in `jit.cyclops`:

1	enables 'edit protect' mode, disabling zone editing
0	disables 'edit protect' mode, enabling zone editing

The following attributes are present in `jit.cyclops`, but not in the original Cyclops object:

<code>dim (get/set)</code>	the dimensions of the object's matrix (similar to <code>capsize</code> in the original object)
<code>planecount (get)</code>	the planecount of the object's matrix
<code>type (get)</code>	the type of the object's matrix (always char)
(others...)	consult the Jitter Manual for other messages common to Jitter objects.

