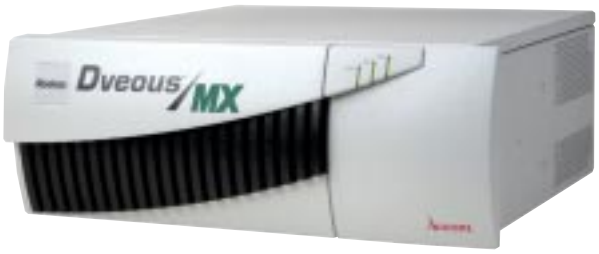
Quick Contents

Section 9: OrbitalFX



Àccom®

Introduction

Dveous/MX achieves its video effects via a collection of parameters, e.g. Locate H and Rotate V, that when altered, modifies the video passing through it. Traditionally, these parameters are altered either manually (using the joystick, keyboard, or softknobs) or as part of keyframe based splining while running an effect.

OrbitalFX™ provides a method for altering Dveous/MX parameters. OrbitalFX uses automatic value generators, called oscillators, which are “patched” to system parameters. Once patched, the parameter is under the oscillator's continuous control; the user controls the oscillator. This can be done in real time while the system is running.

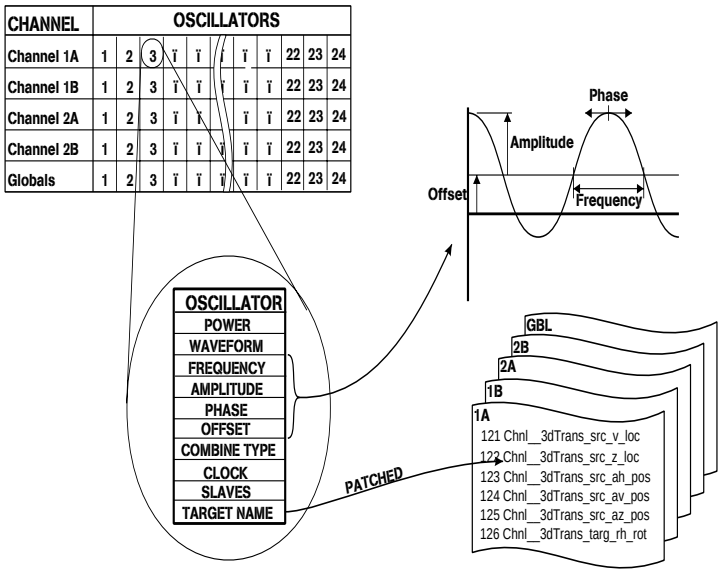
Applying OrbitalFX to multiple parameters (or multiple oscillators to a single parameter) allows you to create complex effects that would have been very difficult or impossible using the traditional keyframe approach. In addition, OrbitalFX is partially integrated into the keyframe based timeline system, allowing you to turn oscillators on and off on a keyframe basis, making it easy to add to keyframe based effects. An example would be to add a subtle but rapid shake to a video compression. Ordinarily this would require many keyframes (depending on the amount of shaking desired); with OrbitalFX it becomes much easier to setup and modify, providing a better chance of getting the look you are after.

Note: State of OrbitalFX can be changed in the Personality menu. See the Technical Guide, P/N 9100-0402-00, or General Operational Notes later in this section for more information.

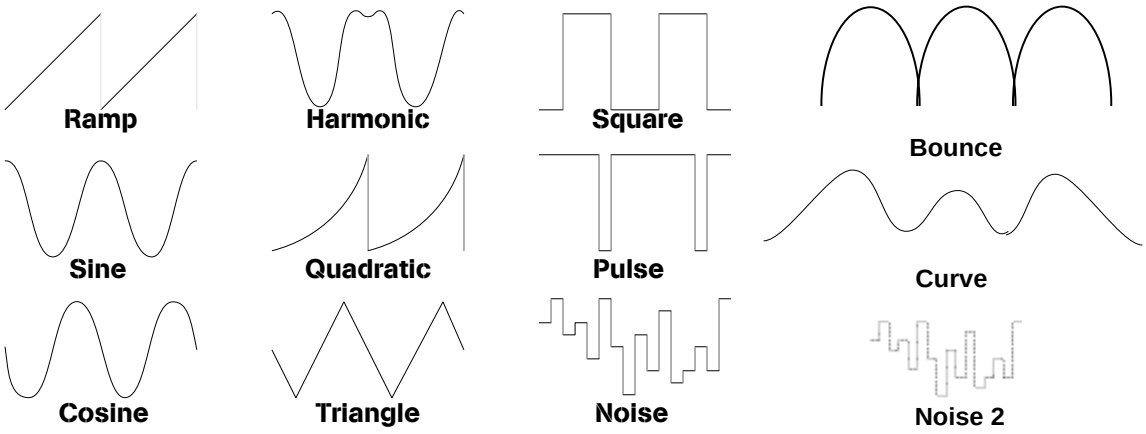
Oscillators

OrbitalFX relies on the use of *oscillators* to generate the values used to modify the various parameters within Dveous/MX. An oscillator is simply an automatic number generator that you control. You set up its amplitude, frequency, the waveform, and its relationship to the other oscillators used.

There are 24 oscillators per channel (1A, 1B, 2A, 2B, and Global) in Dveous/MX. Each comes pre-assigned (patched) to a parameter. You can patch each oscillator to any parameter you like; you can even patch multiple oscillators to the same parameter.



Power turns each oscillator's power on and off. *This field is the only keyframeable parameter*, allowing you to turn on and off individual oscillators within a traditional keyframe timeline.

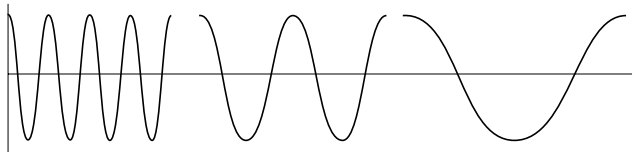


Note: The above waveforms show the output values over time. By following the progression of a waveform, you can see how it controls number generation over time.

- *WAVEFORM* selects the "shape" that the oscillator generates.
  - Ramp: Counts linearly to the maximum amplitude then resets and goes again.
  - Sine: Counts up to the maximum amplitude then counts down to the minimum, following a sine wave shaped curved path. Depending on the usage, this generally gives the appearance of acceleration and deceleration to the movement (when patched to parameters that cause movement) as the limits are reached.
  - Cosine: Operates exactly as with Sine, but 90 degrees out of phase.
  - Harmonic: Counts in a modified sine fashion, but adds a special little harmonic wiggle in the middle.
  - Quadratic: Counts like Ramp, but in a non-linear exponential fashion.
  - Triangle: Counts up to the maximum amplitude then counts down to the minimum (similar to Sine and Cosine) following a linear path.
  - Square: Simply jumps between the maximum and minimum values, spending an equal amount of time at each.

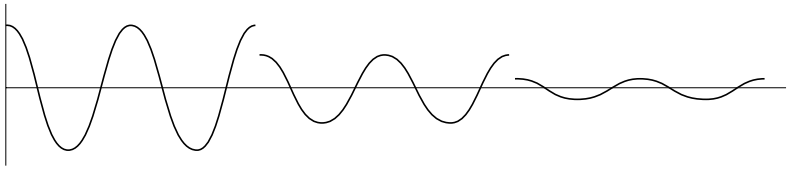
- Pulse: Same as Square, except it spends more time at the maximum value (90%) than at the minimum (10%).
- Noise**:** Randomly generates numbers (within the maximum/minimum range). Note that for this waveform, *smaller* numbers in the ***Frequency*** setting causes random numbers to be generated at a quicker pace.
- Noise 2: Shares a single random number between all oscillators using the NOISE2 wave form for the whole field.
- Bounce: Approximates the acceleration of a bouncing ball.
- Curves: Random but smooth curves.

■ ***Frequency*** determines the speed of the oscillator's counting. Its range is from 0 (stopped) to 99,999 (ridiculously fast). The frequency value represents how much the clock advances per field in the ***Clock*** field. An exception is with the **Noise** waveform, where smaller values produce faster changes.



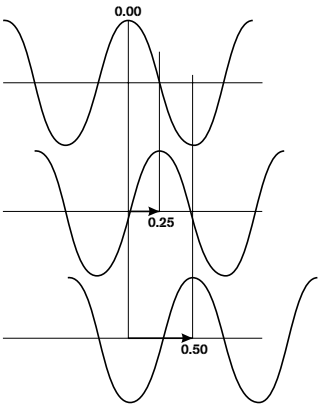
■ ***Amp*** controls the amplitude of the oscillator. It is the distance from the baseline to the maximum and minimum values; the total distance covered is actually twice the ***Amp*** setting. Its range is from –999.9999 to +999.9999. The actual values you use depend on the oscillator's patched parameter. The default ***Amp*** values assigned to each oscillator give a good starting point.

A value of 2.0000 indicates that the oscillator will run from –2.0000 through 0.0000 to +2.0000, an overall range of 4.0000.

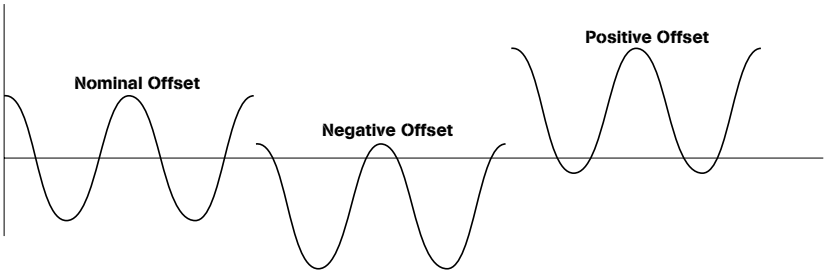


Note: Setting Amp to –2.0000 produces the same results, but the action will be backwards compared to when using a positive number.

- ***Phase*** adds an offset to an oscillator's clock, and can be used to create special relationships between waveforms in related oscillators. For example, two oscillators patched to H Locate on two different channels will appear to follow each other if one is given a small phase adjustment. The ***Phase*** range is from 0.0000 to 1.0000.



- ***Offset*** changes the center point about which the ***Amp*** setting works. In other words, use it to offset the maximum and minimum values from the default position. The range is from –999.9999 to +999.9999, but is dependent on the parameter it is patched to. If ***Amp*** is set to 0, then adjusting ***Offset*** is just like adjusting the parameter itself.



- ***Comb Type*** (Combine Type) selects between *Set*, *Add*, and *Multiply*. When patching multiple oscillators to a single parameter, it is important to know that Dveous/MX applies the oscillators in the order they appear in the center screen display. See Oscillator Combining later for more details.

  Set replaces the parameter's existing values with those its oscillator generates. In this case, you cannot adjust a parameter, manually or with a timeline effect. An oscillator using *Set* overrides any previous ones patched to the same parameter.

  Add trims the parameters values. Use *Add* to apply oscillator generated values to an existing value, whether from a timeline effect or previous oscillators patched to the same parameter.

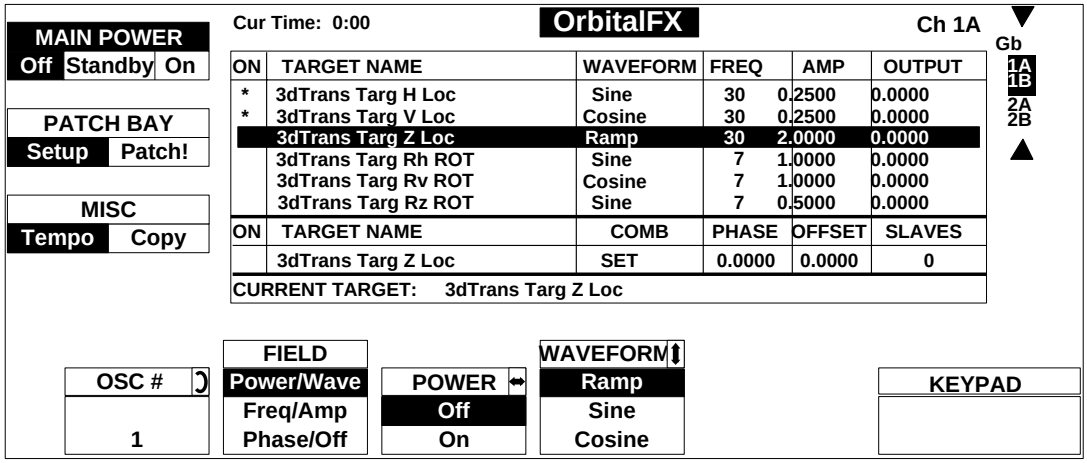
  Multiply can be used to provide a scaling or “volume control” of previous oscillators patched to the same parameter. Using *Multiply* on the first oscillator of a group or when a single oscillator is patched to a parameter causes the parameter's value to go to zero.

- ***Clock*** shows the oscillator's clock. It counts from 0.0000 to 1.0000 at a rate determined by the frequency setting. (The frequency value represents how much the clock advances per field.)
- ***Slaves*** selects the number of oscillators, immediately following the current one (the master), which will respond to changes in the master oscillator. The range is from 0 (no slaves) to 23. For example, if you set ***Slaves*** to 4 on oscillator 6, then oscillators 7 through 10 will all be powered on when oscillator 6 is powered on, and any changes made to 6 will ripple down to the slaves. Once enabled, you can go back and change the settings on any of the slaved oscillators, although any changes made to the master will once again force the slaves to match its settings. The default oscillator patches include slave settings based on parameters. For example, oscillator 1 by default is patched to "3dTrans Targ H Loc", and the following two oscillators (V and Z Loc) slaved to it. Set ***Slaves*** on oscillator 1 to 0 before enabling power on it if you do not want the slaved oscillators to follow.

  The ***Slaves*** setting also affects the Patch Bay mode. Patching an oscillator to a parameter causes any slaved oscillators to be automatically patched to consecutive parameters.

OrbitalFX Menu

Press the **ORBITAL FX** button to see the OrbitalFX menu. (Note: this is the button next to **JOYSTICK LOCK** in the Graphics section of the panel.)



The center of the menu display shows six of the oscillators and their settings, with the currently selected oscillator highlighted.

The *ON* column indicates currently enabled oscillators. *TARGET NAME* shows the functions assigned to each oscillator. A line below the group of six shows more details about the currently selected oscillator. The bottom line shows the currently patched parameter, or in the Patch Bay menu, the parameter to be patched. There are three softkeys along the left side of the display as follows:

- **Main Power**: This allows you to turn the entire OrbitalFX engine on or off, or to place it in a standby mode. It also allows access to the primary oscillator controls, making it the primary menu.
- **Patch Bay**: This allows you to patch (assign) oscillators to the parameters.
- **Misc**: This contains a master tempo adjustment and oscillator copy functions.

Main Power

Toggling **Main Power** *Off* completely disables the OrbitalFX system.

Toggling to *Standby* allows all oscillators set on to connect with their parameters. If you manually adjust their clocks, their output, and thus the parameter value, changes. This is useful when cueing an oscillator to start at a specific point, for example when you first want to control a parameter by timeline control, then switch to oscillator control. To make the transition seamless, note the

parameter's ending value on the timeline, then cue the oscillator to that value by manually adjusting the clock while in Standby mode. When the keyframe is modified or inserted, the clock value sets its starting point.

Toggling **Main Power** *On* engages the OrbitalFX system.

Note: The above settings are not stored as part of the effect.

The softknob settings are as follows:

Osc # Use this to select the oscillator you want to configure. Its number range is from 1 to 24.

Field This actually selects which pairs of oscillator fields the C and D softknobs allow you to modify. See the previous text regarding oscillators for details on the *Field* settings.

- Power/Wave
- Freq/Amp
- Phase/Off
- Comb
- Clock/Slave

Patch Bay

Use this softkey to assign oscillators to the parameters. Softknob A (***Osc #***) selects which oscillator to change, and softknob D (*Var*) selects the parameter you would like to patch to it. *Var*'s range is from 0 to 300, with 0 to 184 actually used to select parameters (0 to 124 with globals). The end of this document contains a list of the parameters and their numbers.

Once you have selected the desired parameter, press the **Patch Bay** softkey again to confirm the patch and assign it.

As mentioned previously, the ***Slaves*** setting also affects the Patch Bay mode. Patching an oscillator to a parameter causes any slaved oscillators to be automatically patched to consecutive parameters.

Misc

The **Misc** Softkey toggles between two modes: Tempo and Copy.

Tempo *Tempo* provides a master clock control, allowing you to speed up or slow down all of the oscillators simultaneously. None of the oscillator values are directly changed by this setting though.

Copy The copy function is useful to rearrange the order of oscillator patches, without having to actually repatch them. This is useful when you need to alter the parameter hierarchy for an effect.

- Store Osc—***Store Osc*** copies the current oscillator to another. Type the destination's number at the numeric keypad and press the **A** button.
- Recall—***Recall*** copies an oscillator into the current location. Type the source's number at the numeric keypad and press the **B** button.

Note: The A and B softknobs do not function in this mode; the values of "–1" have no meaning.

OrbitalFX and the Keyframe System

Power (on/off control) for each oscillator is the only OrbitalFX parameter changeable on a keyframe basis within an effect. The status of the other parameters is remembered, but cannot be splined (changed) within an effect.

Beware of having oscillators running while modifying or inserting keyframes, as you could end up with a basic effect containing undesired OrbitalFX influences. The safest approach is to build the basic effect first, then with the OrbitalFX **Main Power** off, configure and enable the desired oscillators, modifying the appropriate keyframes where each oscillator's power is turned on or off. All OrbitalFX settings are put into workspace once you modify or insert a keyframe after it is configured. At this point it is safe to enable OrbitalFX **Main Power**.

Note: The status of Main Power is not a part of the effect, and is not remembered when saving it.

If you change an oscillator's setting, for example its frequency, you can modify any keyframe to write the change to the workspace effect.

Using OrbitalFX with Keyframe Effects

Use care when patching oscillators to parameters used within an effect. The result could be a subtle modification or a dramatic change.

The **Comb** field setting determines whether the output of a given oscillator replaces the value for its parameter (*Set*), or just trims it (*Add*). You cannot adjust any parameter with an oscillator patched to it that has the Comb field to *Set*, manually or with an effect – OrbitalFX has total control over that parameter if the oscillator is powered on and Main Power is set On or Standby.

Oscillator Combining

Applying multiple oscillators to a single parameter leads to a wide variety of results, from adding subtle harmonic effects to ramping oscillator effects on and off. The Comb Type (Combine Type) setting determines how the oscillators interact.

Dveous/MX effectively creates a mathematical expression from the oscillator settings to determine the action of the parameter. The expression is determined by oscillator order. It is important to take this into account when setting up your effect.

Comb Type (Combine Type) selects between *Set*, *Add*, and *Multiply*.

- *Set* replaces the parameter's existing values with those its oscillator generates. An oscillator using *Set* overrides any previous ones patched to the same parameter.
- *Add* trims the parameters values. An oscillator using *Add* has its values added to any previous ones patched to the same parameter.
- *Multiply* provides a scaling or "volume control" of previous oscillators patched to the same parameter. It is not advised to use *Multiply* on the first oscillator of a group.

Combining Example

The following example uses 3d Trans Target H Locate, and assumes Dveous/MX is at the factory normal.

1. Set **3D Trans** *Target Z Locate* to 56.
2. In the OrbitalFX menu, set **Main Power** *On*.
3. Select oscillator 1. It should already be patched to 3dTrans Targ H Loc.
4. Set **Slaves** to *0*.
5. Set **Power** *On*, **Waveform** to *Sine*, **Freq** to *100,* and **Amp** to *3.0*. You should now see the picture moving back and forth across the screen.
6. Toggle the **Misc** softkey to select *Copy*. Press *2* then **A** (Store Osc). This makes a copy of the current oscillator (1) and puts it into oscillator 2.
7. Press **Main Power** then select oscillator 2. Turn power on, set its **Comb** to *Add*, **Freq** to *900*, and **Amp** to *0.25*. The picture should now "stutter" as it moves back and forth. This is the second oscillator being added to the original.

8. Toggle the **Misc** softkey to select *Copy*. Press *3* then **A** (Store Osc). This makes a copy of the current oscillator (2) and puts it into oscillator 3.
9. Press **Main Power** then select oscillator 3. Turn power on, set its **Comb** to *Multiply*, **Waveform** to *Ramp*, **Freq** to *10*, **Amp** to *0.5*, and **Offset** to *0.5*. The picture should now start with a small back and forth movement in the center that gradually grows. Once it reaches its limit it jumps back to being a small movement.

The above example sets up osc 1 + osc 2 x osc 3. Setting oscillator 3's Amp to 0.5 gives it a total swing of 1.0 (0.5 to –0.5); the Offset of 0.5 ensures the ramp goes from 0.0 to 1.0. Multiplying the result of osc 1 + osc 2 by 0.0 (the bottom of the ramp) gives a result of 0.0; multiplying by 1.0 (the top of the ramp) gives the result of full movement. The ramp values between 0.0 and 1.0 produce scaled results.

To use the above to turn up the action and leave it up, add a keyframe that turns oscillator 3's power off. The duration of keyframe one (which controls how long oscillator 3 is on) needs to be just long enough for the multiply to get to its end. Try different settings until you get the proper result.

Aligning Oscillator Clocks

Each oscillator's clock starts running when its power is turned on. If clocks of related or synchronizing oscillators get out of alignment, align them in one of two ways.

To align the clocks of all oscillators on delegated channels, press the **ALIGN** + **ORBITAL FX** buttons at the same time.

Align the current oscillator and its slaves (on all delegated channels) by pressing **ALIGN** + **A** (*OSC #*). Simply twisting the **Clock** softknob in that situation also aligns the appropriate clocks, setting the their clocks to the same value, whatever that may be. Use the numeric keypad to enter a specific value and align the clocks.

General Operational Notes

- Use **CLEAR** + **ORBITAL FX** to set all oscillators to default (on all delegated channels).
- Use **CLEAR** + **A** (*Osc #*) to set the current oscillator and its slaves to default (on all delegated channels).

- RUN ENV in the PERSONALITY Menu under MISC, allows you to select how OrbitalFX will run.
  - Post - Recall the saved OrbitalFX state when you run.
  - Live - DON'T recall the saved state. This allows the live user to have an OrbitalFX effect running and hit RUN to start a timeline portion of the effect. The OrbitalFX clocks will not reset and cause a hiccup.

Patch Bay *Var* Setting

There are two concerns when patching oscillators to parameters from the list below: figuring out the real name for the parameter you want to patch, and knowing the proper ***Amp*** and ***Offset*** values.

The parameters vary greatly on their number requirements, and a lot of them do not even use numbers (on and off settings and such). Set ***Amp*** with the range of the given parameter in mind. To control a parameter that has an on and off setting, set ***Amp*** to 1 and ***Offset*** to 1. For a parameter that has 4 settings (*1 Chnl_Input_front_inv* for example), set ***Amp*** to 2 and ***Offset*** to 2. Remember that the ***Amp*** value determines how far above and below the ***Offset*** value the oscillator goes.

Several of these parameters do not lend themselves to OrbitalFX control (for example *73 Chnl_warp_shape*), and produce results that are mostly useless.

Note: Most of the following parameters can be figured out with their name. Those that need extra explanation have italics text in parenthesis added.

Local Channel Parameters

3D TRANS
117 Chnl__3dTrans_src_rh_rot
118 Chnl__3dTrans_src_rv_rot
119 Chnl__3dTrans_src_rz_rot
120 Chnl__3dTrans_src_h_loc
121 Chnl__3dTrans_src_v_loc
122 Chnl__3dTrans_src_z_loc
123 Chnl__3dTrans_src_ah_pos *(axis)*
124 Chnl__3dTrans_src_av_pos *(axis)*
125 Chnl__3dTrans_src_az_pos *(axis)*
126 Chnl__3dTrans_targ_rh_rot
127 Chnl__3dTrans_targ_rv_rot
128 Chnl__3dTrans_targ_rz_rot
129 Chnl__3dTrans_targ_h_loc
130 Chnl__3dTrans_targ_v_loc
131 Chnl__3dTrans_targ_z_loc
132 Chnl__3dTrans_targ_ah_pos *(axis)*
133 Chnl__3dTrans_targ_av_pos *(axis)*
134 Chnl__3dTrans_targ_az_pos *(axis)*

2D TRANS
135 Chnl_Trans_h_pan *(post H)*
136 Chnl_Trans_v_pan *(post V)*
137 Chnl_Trans_zoom *(post Zoom)*
138 Chnl_Trans_h_inpos *(pre H)*
139 Chnl_Trans_v_inpos *(pre V)*
140 Chnl_Trans_h_aspect
141 Chnl_Trans_v_aspect
142 Chnl_Trans_src_size *(pre/aspect size)*
143 Chnl_Trans_h_skew
144 Chnl_Trans_v_skew
145 Chnl_Trans_perspcv
BORDER CROP
30 Chnl_Crop_cpin_left
31 Chnl_Crop_cpin_top
32 Chnl_Crop_cpin_right
33 Chnl_Crop_cpin_bottom
49 Chnl_Crop_cpin_soft_left
50 Chnl_Crop_cpin_soft_top
51 Chnl_Crop_cpin_soft_right
52 Chnl_Crop_cpin_soft_bottom
55 Chnl_Border_bottom_width
56 Chnl_Border_left_width
57 Chnl_Border_right_width
58 Chnl_Border_top_width
59 Chnl_Border_opac
60 Chnl_Border_hue
61 Chnl_Border_lum
62 Chnl_Border_sat
63 Chnl_Border_softness
MULTI
22 Chnl_Mosaic_h_tile
23 Chnl_Mosaic_v_tile
24 Chnl_Mosaic_size
25 Chnl_Mosaic_mosaic_on
29 Chnl_Mosaic_motdet
42 Chnl_Crop_cpin_h_multi
43 Chnl_Crop_cpin_v_multi
COLOR CRCTR
146 Chnl_Ccr_Y_Gain
147 Chnl_Ccr_U_Gain
148 Chnl_Ccr_V_Gain
149 Chnl_Ccr_Y_Offset
150 Chnl_Ccr_U_Offset
151 Chnl_Ccr_V_Offset
152 Chnl_Ccr_Y_Gamma
153 Chnl_Ccr_U_Gamma
154 Chnl_Ccr_V_Gamma
155 Chnl_Ccr_Y_Knee
156 Chnl_Ccr_U_Knee
157 Chnl_Ccr_V_Knee

158 Chnl_Ccr_Proc_Hue
159 Chnl_Ccr_Proc_Sat
160 Chnl_Ccr_Proc_Gain
161 Chnl_Ccr_Proc_Offset
176 Chnl_Ccr_YuvGbrMode

COLOR MODIFY

162 Chnl_Ccr_Y_Sol
163 Chnl_Ccr_U_Sol
164 Chnl_Ccr_V_Sol
165 Chnl_Ccr_Tint_Enable
166 Chnl_Ccr_Black_Sat
167 Chnl_Ccr_Black_Hue
168 Chnl_Ccr_Mid_Sat
169 Chnl_Ccr_Mid_Hue
170 Chnl_Ccr_White_Sat
171 Chnl_Ccr_White_Hue
172 Chnl_Ccr_Invert_Gain
173 Chnl_Ccr_Y_Inv_Gain
174 Chnl_Ccr_U_Inv_Gain
175 Chnl_Ccr_V_Inv_Gain

INPUT

1 Chnl_Input_front_inv
2 Chnl_Input_back_inv
3 Chnl_Input_front_type
4 Chnl_Input_front
5 Chnl_Input_back_type
6 Chnl_Input_back
45 Chnl_Crop_cpin_blur (V blur)
46 Chnl_Crop_cpin_h_blur
181 Chnl_Freeze_mode
182 Chnl_Freeze_type
183 Chnl_Freeze_strobe
184 Chnl_Freeze_duty

WARP

64 Chnl_warp_f__mod
(V Freq, Split V Position Bottom Right)
65 Chnl_warp_decay
(Split V Position Bottom Left)
66 Chnl_warp_h_pos
(Position, H Position, H Split)
67 Chnl_warp_v_pos
(V Position, V Split)
68 Chnl_warp_spread
(Frequency, H Freq, Radius, Size, Spread, Split V Position Top Right)
69 Chnl_warp_range
(Amplitude, Mag, Range, Twist, Split V Position Top Left)
70 Chnl_warp_axis
(Axis, Rotation, Split Rotate Top Left)
71 Chnl_warp_phase

(Alignment, Phase, Split Rotate Top Right)
72 Chnl_warp_p__width (Pulsewidth)
73 Chnl_warp_shape
75 Chnl_warp_quadrant_1_offset *(Cylinder Position, Left, Ring 1, Top Left,* Split H Position Top Left)
76 Chnl_warp_quadrant_2_offset (Right, Ring 2, Top Right, Split H Position Top Right)
77 Chnl_warp_quadrant_3_offset *(Bottom Left, Ring 3, Top, Split H* Position Bottom Right)
78 Chnl_warp_quadrant_4_offset *(Bottom, Bottom Right, Ring 4, Split H* Position Bottom Left)
79 Chnl_warp_useful_one (Split Rotate Bottom Right)
80 Chnl_warp_useful_two (Split Rotate Bottom Left)

LIGHT SOURCE

81 Chnl_Light_src_xOver *(Source)*
82 Chnl_Light_src_gainMode *(Model)*
83 Chnl_Light_src_hiLoEnable *(Lights)*
84 Chnl_Light_src_hiHue *(Specular)*
85 Chnl_Light_src_hiLum *(Specular)*
86 Chnl_Light_src_hiSat *(Specular)*
87 Chnl_Light_src_hiOpac *(Specular)*
88 Chnl_Light_src_loHue *(Specular)*
89 Chnl_Light_src_loLum *(Specular)*
90 Chnl_Light_src_loSat *(Specular)*
91 Chnl_Light_src_loOpac *(Specular)*
92 Chnl_Light_src_hiYGain *(Diffuse)*
93 Chnl_Light_src_hiCGain *(Diffuse)*
94 Chnl_Light_src_loYGain *(Diffuse)*
95 Chnl_Light_src_loCGain *(Diffuse)*
96 Chnl_Light_src_ambYGain *(Diffuse)*
97 Chnl_Light_src_ambCGain *(Diffuse)*
99 Chnl_Light_src_type *(Spot/Bar/Flood)*
100 Chnl_Light_src_zpos
101 Chnl_Light_src_vpos
102 Chnl_Light_src_hpos
103 Chnl_Light_src_hrot *(Bar)*
104 Chnl_Light_src_vrot *(Bar)*
105 Chnl_Light_src_zrot *(Bar)*
116 Chnl_Light_src_gamma *(Focus)*
180 Chnl_Light_src_mode

TEXTURE

106 Chnl_Light_src_cropEnable
107 Chnl_Light_src_cropLeft

108 Chnl_Light_src_cropRight
109 Chnl_Light_src_cropTop
110 Chnl_Light_src_cropBottom
111 Chnl_Light_src_txtOffset
112 Chnl_Light_src_txtGain
113 Chnl_Light_src_texture *(Main-Source)*
114 Chnl_Light_src_txtFreeze
115 Chnl_Light_src_txtEnable

KEY

0 Chnl_Operating_mode *(VV/VK/VKS)*
7 Chnl_Input_keyStat *(White/Video)*
8 Chnl_Input_shadStat *(White/Video)*
12 Chnl_Keyer_setup_gain
13 Chnl_Keyer_setup_clip
14 Chnl_Keyer_setup_phase
15 Chnl_Keyer_setup_hue *(Shadow)*
16 Chnl_Keyer_setup_lum *(Shadow)*
17 Chnl_Keyer_setup_sat *(Shadow)*

CORNER PIN

34 Chnl_Crop_cpin_h_corner *(Top Left)*
35 Chnl_Crop_cpin_h_corner2 *(Top Right)*
36 Chnl_Crop_cpin_h_corner3 *(Bottom Rt)*
37 Chnl_Crop_cpin_h_corner4 *(Bottom Left)*
38 Chnl_Crop_cpin_v_corner *(Top Left)*
39 Chnl_Crop_cpin_v_corner2 *(Top Right)*
40 Chnl_Crop_cpin_v_corner3 *(Bottom Rt)*
41 Chnl_Crop_cpin_v_corner4 *(Bottom Left)*
47 Chnl_Crop_cpin_autokey
48 Chnl_Crop_cpin_cmotion

OUTPUT

9 Chnl_Keyer_setup_a_opac
10 Chnl_Keyer_setup_b_opac
11 Chnl_Keyer_setup_cross_fade
18 Chnl_Keyer_setup_priority
19 Chnl_Keyer_setup_z_offset_a
20 Chnl_Keyer_setup_z_offset_b
21 Chnl_Keyer_setup_z_softness

LOCAL NOT SUPPORTED

26 Chnl_Mosaic_image
27 Chnl_Mosaic_hasp
28 Chnl_Mosaic_vasp
44 Chnl_Crop_cpin_select
53 Chnl_Crop_cpin_key_source
54 Chnl_Crop_cpin_key_screen
74 Chnl_warp_wave
98 Chnl_Light_src_blackLevel
177 Chnl_Input_front_Ext_Xpnt
178 Chnl_Input_back_Ext_Xpnt
179 Chnl_Pause_On

SPARES (UNUSED)

185 Chnl_Spare1

186 Chnl_Spare2
187 Chnl_Spare3
188 Chnl_Spare4
189 Chnl_Spare5
190 Chnl_Spare6

Global Channel Parameters

GLOBAL TRANS

0 Gbl__3dTrans_src_rh_rot
1 Gbl__3dTrans_src_rv_rot
2 Gbl__3dTrans_src_rz_rot
3 Gbl__3dTrans_src_h_loc
4 Gbl__3dTrans_src_v_loc
5 Gbl__3dTrans_src_z_loc
6 Gbl__3dTrans_src_ah_pos *(axis)*
7 Gbl__3dTrans_src_av_pos *(axis)*
8 Gbl__3dTrans_src_az_pos *(axis)*
9 Gbl__3dTrans_targ_rh_rot
10 Gbl__3dTrans_targ_rv_rot
11 Gbl__3dTrans_targ_rz_rot
12 Gbl__3dTrans_targ_h_loc
13 Gbl__3dTrans_targ_v_loc
14 Gbl__3dTrans_targ_z_loc
15 Gbl__3dTrans_targ_ah_pos *(axis)*
16 Gbl__3dTrans_targ_av_pos *(axis)*
17 Gbl__3dTrans_targ_az_pos *(axis)*
18 Gbl__2dTrans_h_pan *(no keyboard equiv)*
19 Gbl__2dTrans_v_pan *(no keyboard equiv)*
20 Gbl__2dTrans_zoom *(no keyboard equiv)*

COMBINER

37 Gbl_Dve_combiner_cross_fade
38 Gbl_Dve_combiner_ab_opac *(ch1)*
39 Gbl_Dve_combiner_cd_opac *(ch2)*
40 Gbl_Dve_combiner_priority *(fixed/zkey)*
41 Gbl_Dve_combiner_z_softness
42 Gbl_Out_combiner_cross_fade *(TFS)*
43 Gbl_Out_combiner_dve_opac *(TFS)*
44 Gbl_Out_combiner_tfs_opac *(TFS)*
45 Gbl_Out_combiner_priority *(TFS fix/zkey)*
46 Gbl_Out_combiner_z_softness *(TFS)*
48 Gbl_Background_source
49 Gbl_Background_freeze
76 Gbl_Background_z_pos
77 Gbl_Background_z_softness
78 Gbl_Background_z_cross_fade
79 Gbl_Background_priority *(fixed/zkey)*
80 Gbl_Framestore_source *(TFS source)*

TARGET FRAMESTORE

47 Gbl_Framestore_freeze *(TFS)*
82 Gbl_tfs_decay
83 Gbl_tfs_slinky
84 Gbl_tfs_sparkle_size

85 Gbl_tfs_sparkle_rate
86 Gbl_tfs_hue
87 Gbl_tfs_sat
88 Gbl_tfs_lum
89 Gbl_tfs_z_soft *(Comp mode)*
90 Gbl_tfs_xfade *(Comp mode)*
91 Gbl_tfs_effect_offOn
92 Gbl_tfs_effect_mode *(Attributes)*
94 Gbl_tfs_output_mode
96 Gbl_tfs_decay_offOn
97 Gbl_tfs_sparkle_offOn
98 Gbl_tfs_drop_offOn
99 Gbl_tfs_xfade_mode
100 Gbl_tfs_trail_color_offOn

DEFOCUS

21 Gbl_Defocus_one_src
22 Gbl_Defocus_one_aperature
23 Gbl_Defocus_one_luma_h
24 Gbl_Defocus_one_luma_v
25 Gbl_Defocus_one_luma_enable
26 Gbl_Defocus_one_chroma_h
27 Gbl_Defocus_one_chroma_v
28 Gbl_Defocus_one_chroma_enable
29 Gbl_Defocus_two_src
30 Gbl_Defocus_two_aperature
31 Gbl_Defocus_two_luma_h
32 Gbl_Defocus_two_luma_v
33 Gbl_Defocus_two_luma_enable
34 Gbl_Defocus_two_chroma_h
35 Gbl_Defocus_two_chroma_v
36 Gbl_Defocus_two_chroma_enable

SUPERMATTE

50 Gbl_SMatte_pattern
51 Gbl_SMatte_rot_mode *(angle/spin)*
52 Gbl_SMatte_rot_angle
53 Gbl_SMatte_rot_spin
54 Gbl_SMatte_split_enable
55 Gbl_SMatte_split_level
56 Gbl_SMatte_h_multiple
57 Gbl_SMatte_v_multiple
58 Gbl_SMatte_reflect
59 Gbl_SMatte_progress
60 Gbl_SMatte_hpos
61 Gbl_SMatte_vpos
62 Gbl_SMatte_softness
63 Gbl_SMatte_pers_mag
64 Gbl_SMatte_pers_rot
65 Gbl_SMatte_colour1_hue
66 Gbl_SMatte_colour1_lum
67 Gbl_SMatte_colour1_sat
68 Gbl_SMatte_colour2_hue

69 Gbl_SMatte_colour2_lum
70 Gbl_SMatte_colour2_sat
71 Gbl_SMatte_aspect
72 Gbl_SMatte_rings1
73 Gbl_SMatte_arms1
74 Gbl_SMatte_rings2
75 Gbl_SMatte_arms2

SOLID BUILDER

101 Gbl_Slab_enable
102 Gbl_Slab_width
103 Gbl_Slab_height
104 Gbl_Slab_depth
105 Gbl_Slab_frontPosV
106 Gbl_Slab_frontSizeV
107 Gbl_Slab_frontPosH
108 Gbl_Slab_frontSizeH
109 Gbl_Slab_topPosV
110 Gbl_Slab_topSizeV
111 Gbl_Slab_topPosH
112 Gbl_Slab_topSizeH
113 Gbl_Slab_sidePosV
114 Gbl_Slab_sideSizeV
115 Gbl_Slab_sidePosH
116 Gbl_Slab_sideSizeH
117 Gbl_Slab_slabSide
118 Gbl_Slab_slabSide1B
119 Gbl_Slab_slabRoll_Dir
120 Gbl_Slab_enable_ch1a
121 Gbl_Slab_enable_ch1b
122 Gbl_Slab_enable_ch2a
123 Gbl_Slab_enable_ch2b

GLOBAL NOT SUPPORTED

77 Gbl_Framestore_z_enable
89 Gbl_tfs_source
91 Gbl_tfs_blank_mode

Warp Menu Parameters

The challenge in controlling the Warp menu parameters is determining which patches apply to each warp shape's controls. The following list provides *Var* values assigned to the *Modify* and *Position* controls used by the warp shapes. Warp pattern #71, Split, has its *Position* controls listed separately at the end.

The number (68 for Spread for example) is the ***VAR*** value to enter in the Patch Bay mode. The Warp View controls have no tie to OrbitalFX (and would be peculiar to use if they did).

MODIFY

Alignment	71
Amplitude	69
Axis	70
Frequency	68
H Frequency	68
Mag	69
Phase	71
Position	75
Pulsewidth	72
Radius	68
Range	69
Rotation	70
Size	68
Spread	68
Twist	69
V Frequency	64

POSITION

Bottom	78
Bottom Left	77
Bottom Right	78
H Position	66
H Split	66
Left	75
Position	66
Right	76
Ring 1	75
Ring 2	76
Ring 3	77
Ring 4	78
Top	77
Top Left	75
Top Right	76
V Position	67
V Split	67

SPLIT V POSITION

Bottom Right	64
Bottom Left	65
Top Right	68
Top Left	69

SPLIT H POSITION

Bottom Right	77
Bottom Left	78
Top Right	76
Top Left	75

SPLIT ROTATE

Bottom Right	79
Bottom Left	80
Top Right	71
Top Left	70

Continuous Rippling Effect

This section describes creating an effect that causes the movement to appear to run continuously. The concept can be applied to numerous situations, but the example used is a common one involving circular ripples. The goal is to make the image look like the rippling never stops.

The basic task is to have OrbitalFX change the Warp menu's Phase parameter so that a ripple moves exactly one ripple period, as determined by the Warp menu's Frequency setting, then jump back to where it started and go again. The OrbitalFX Ramp waveform is perfect for actually running the effect, but is less than ideal for setting it up. For that we will use the Square waveform.

The following assumes you have good video for rippling purposes (flat color fields are disappointing). The grid test pattern of the Texture menu works very well.

1. Press **NORMAL NORMAL NORMAL ENTER** to force Dveous/MX to a known starting point.
2. Press the **WARP** button, set **Warp** *On*, **Category** to *Ripple*, and **Type** to *Circular*. Toggle the **Demo** mode *On* then *Off*. This applies static ripples to the picture.
3. Press the **ORBITALFX** button. Ensure oscillator 1 is selected, and set its **Slaves** to *0*.
4. Use **Patch Bay** to patch VAR *70* to oscillator 1.
5. Set the oscillator's *Power* to *On*, **Waveform** to *Square*, and **Frequency** to *200*.
6. Start turning up the amplitude. You will see a rapid cut between the amplitude limits. As you keep turning up the amplitude you will see that there is a point where the two images appear as one. This indicates that amplitude is jumping the Warp Phase parameter exactly one ripple period. (The amplitude value should be about 0.25 in this example; this value is determined by the Warp Frequency setting.)
7. Change the oscillator's **Waveform** to *Ramp*. The ripples now appear to be perpetually moving from the center out.