Using Formulas in Ray Dream Studio

Ray Dream allows you to create mathematical formulas that in turn create custom features. Using formulas, you can define your own objects, deformers, tweeners, shaders, gels, backgrounds, backdrops.

The interface for creating formulas is common for all items. The valid input and output variables differ for each item you create.

Formula editing becomes technical quickly. Higher math or programming experience is recommended.

This document describes the following:

- How to use a formula item
- The Formula Editor interface
- Supported functions, operators, constants and parameters
- The valid input and output variables for each formula item.

Creating Formula Items

The method for creating a formula item depends on the item you want to create. You can create the following Ray Dream Studio items using formulas:

- Primitve objects
- Deformers
- Tweeners
- Shaders
- Gels
- Backgrounds and Backdrops

To create a Formula primitve:

1. Drag the Formula Primitive icon into the scene.
2. Jump Into the object to display the formula controls.

To create a Formula deformer:

1. Display the **Properties palette: Deformer tab**.
2. Choose **Formula** from the pop-up. The formula controls appear in the Deformer tab.

To create Formula tweeners:

1. Double-click the gap between key events on the time track. Studio opens the **Transition Options** dialog.
2. Choose **Formula** from the pop-up. The formula controls appear in the **Transition Options** dialog.

To create Formula shaders:

1. Display the **Current Shader Editor** by double-clicking a shader in the **Browser**.
2. Select the shader tree component where you want to use a formula.
3. Choose **Current Shader Editor palette: Insert menu▸ Formula**. The formula controls appear on the selected branch of the shader tree.

To create Formula gels:

1. Select a light.
2. Display the **Properties palette: Gel tab**.
3. Choose **Formula** from the pop-up. The formula controls appear in the **Gel** tab.

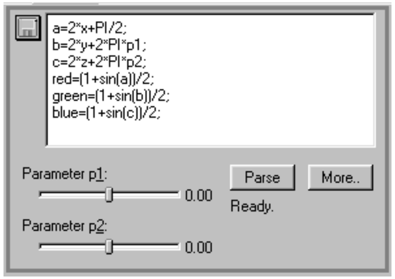
To create Formula backgrounds or backdrops:

1 Display the **Scene Settings window: Effects tab**.

2 From the (Background or Backdrop) pop-up, choose **Formula**. The formula controls appear in the **Effects** tab in the Background or Backdrop controls.

Formula Controls

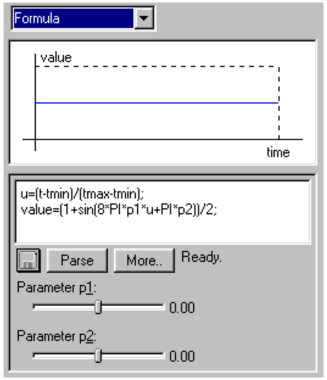
The formula controls include a readout of the formula itself, sliders for adjusting parameters, a **Parse** button, and a **More** button.



```
a=2*x+PI/2;
b=2*y+2*PI*p1;
c=2*z+2*PI*p2;
red=(1+sin(a))/2;
green=(1+sin(b))/2;
blue=(1+sin(c))/2;
```

Use the Formula Editor controls to enter formulas and set variables.

For a few items, the controls also include a preview of sorts. (For the tweener formula, the preview is a graph.)



```
u=(t-tmin)/(tmax-tmin);
value=(1+sin(8*PI*p1*u+PI*p2))/2;
```

The formula editor for tweeners displays a preview of the formula as a graph.

The disk icon is a pop-up that offers Open and Save As commands; so you can create a formula, save it to disk, then open and use it later.

Editing Formulas

To create or edit a formula:

1 In the formula controls, click **More**.

Ray Dream opens the **Formula Editor**.

- You can type into the window to build your formula. Be careful, the editor is case-sensitive.
- You may also use the Operators, Input Variables and Output Variables pop-ups to insert any of these items. Descriptions of the operators and variables follow.
- Each statement must end with a semicolon.

2 Use the disk icon pop-up to save or open a formula.

3 When you're finished in the **Formula Editor**, click **OK**.

4 In the formula controls, click the **Parse** button.

Parsing compiles the formula to run.

5 If appropriate to this formula, you can adjust the parameter sliders.

You can now try out the formula.

Formula Operators

sin(x)	sine of x
cos(x)	cosine of x
tan(x)	tangent of x
asin(x)	arcsine of x

acos(x)	arccosine of x
atan(x)	arctangent of x
atan2(y,x)	arctangent of y/x
sinh(x)	hyperbolic sine of x
cosh(x)	hyperbolic cosine of x
tanh(x)	hyperbolic tangent of x
sqrt(x)	square root of x
pow(x,y)	x raised to the power of y
exp(x)	exponential
log(x)	logarithm
log10(x)	base-10 logarithm
floor(x)	floor of x (The largest integer that is less than or equal to x)
ceil(x)	ceiling of x (The smallest integer that is greater than or equal to x)
abs(x)	absolute value of x
mod(x,y)	x modulus y

Comparison Operators:

<	less than
<=	less than or equal to
==	equal to
!=	not equal to
>	greater than
>=	greater than or equal to
&&	logical "and"
||	logical "or"

Conditional:

variable=((comparison)?(value if TRUE):(value if FALSE))

For example:

a=((b>1)?1:b);

The expression (b>1) is the comparison. If it's true, the formula returns a=1. If it's false, the formula returns a=b.

Constants:

PI

E

Animatable input parameters:

p1, p2, p3, p4

These parameters can be set between -1 and +1

Variables

The variables valid for a particular item depend on its type. Refer to the section below that describes the variables for the formula item you are creating. To define a variable, just use it. Its initial value will be 0.

Formula Shader

Output Variables

value (output is a grey level) or red, green, blue (output is a color)

The expected output is between 0 (black) and 1 (white or color maximum intensity).

You can use any of these output variables, however the program can't use 'value' and any of 'red', 'green', 'blue' simultaneously. If the shader is used in a color based channel, then color output variables will have priority (meaning if 'value' and 'red' are defined, then only 'red' will be used). The opposite applies when using the shader in a value based channel, like Shininess. (If 'value' is defined, 'value' will be used).

Input Variables

u and v	These are parametric mapping coordinates.
x,y and z	These are the local space coordinates of the point being shaded.
nx,ny and nz	These are the normal local coordinates (normalized vector).

X,Y and Z	These are the global space coordinates of the point being shaded.
NX,NY and NZ	These are the normal global coordinates (normalized vector).

Formula Deformer

The Formula Deformer iterates on all the points of an object, deforming them one by one.

Output Variables

dx,dy and dz	The local coordinates of the deformed point. The range is -1 to +1. (0 is the center.)

Input Variables:

x, y, z	The local coordinates of the point to be deformed.

Formula Tweener

Output Variables

value

Input Variables

t, tmin, tmax	t is time. One t unit is 1/1800th of a second. t is between tmin and tmax

Formula Backdrop

Output Variables

value or red, green, blue

Note: See the Formula Shader.

Input Variables

u,v	(u,v) defines a point on the backdrop. The range is between 0 and 1. (0,0) is the lower left corner, (1,1) the upper right.

Formula Background

Output Variables

value or red,green,blue

Note: See the Formula Shader.

Input Variables

x,y,z	(x,y,z) isthe global coordinates of the direction vector, from the camera eye to the background. The vector (x,y,z) is normalized and each coordinate ranges between -1 and +1.

Formula Gel

Output Variables

value or red,green,blue

Note: See the Formula Shader.

Input Variables

u,v	(u,v) defines a point on the backdrop. The range is between 0 and 1. (0,0) is the lower left corner, (1,1) the upper right.

Formula Primitive (Object)

Output Variables

x,y,z	local coordinates of the point defined by (u,v)

Input Variables

u,v	The parametric space coordinates. The range is between 0 and 1