

real

COLLABORATORS

	TITLE : real		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		July 25, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	real	1
1.1	real.guide	1
1.2	keybindings	1
1.3	keybindings2	5
1.4	tags	8
1.5	materials	10
1.6	material properties	10
1.7	material handlers	13
1.8	mapping	13
1.9	scope	13
1.10	bump	14
1.11	color	14
1.12	indexh	15
1.13	material variables	15
1.14	procmat	17
1.15	animation system	18
1.16	built-in methods	18
1.17	path	19
1.18	rotation	19
1.19	sweep	20
1.20	size	20
1.21	stretch	21
1.22	direction	21
1.23	move & dir	22
1.24	control curves	22
1.25	simple skeleton	23
1.26	skeleton	23
1.27	inv kinematic	24
1.28	morphing open	25
1.29	morphing closed	25

1.30	transform	25
1.31	wave	26
1.32	radial force	26
1.33	directed force	27
1.34	tangent force	28
1.35	collision	29
1.36	int collision	29
1.37	friction	30
1.38	creation	31
1.39	processor	31
1.40	rpl	32
1.41	noise	32
1.42	attributes	33
1.43	custom methods	33
1.44	abs path	34
1.45	chain	34
1.46	weird force	35
1.47	evaluatable objects	35
1.48	rpl variables	35
1.49	rendering	36

Chapter 1

real

1.1 real.guide

REAL 3D ON-LINE HELP

Readme

Information not included in the manual	New features, changes
Errors in the manual	Corrections & notes

Real 3D On-Line Help

Default Key Bindings	RPL startup definitions
Tags	Object Tag Descriptions
Materials	Real 3D Material Editor
Rendering Settings	Rendering Window Options
Animation System	Built-in and Custom Methods
Real 3D Programming Language	RPL detailed Information
Custom Tools	Special Tools using RPL

Developer Information

Real 3D Binary File Format Description	3rd Party Programmer Support
Real 3D Display Driver Interface	3rd Party Programmer Support

Real 3D Index

@{ " Index " Link "r3d2:help/index.guide/main"} reference	A-Z all subject ↔
--	-------------------

1.2 keybindings

DEFAULT KEYBINDINGS - by category ALSO SEE by key

Animation

<	< = Play animation backward
>	> = Play animation forward
?	? = Go to
[[= Go to the beginning
]] = Go to the end
b	b = Step Backward 1 frame
f	f = Step Forward one frame
u	u = Refresh = Update Animation System

Boolean Operations

&	& = Boolean AND
%	% = Boolean AND NOT
A	SHIFT A = Boolean AND With Paint
B	SHIFT B = Boolean OR
N	SHIFT N = Boolean AND NOT With Paint
w	w = Rethink Wireframe
W	SHIFT W = Rethink Wireframe all sublevels

Creation

g	g = Create subGroup object
h	h = Create methOd level = Animation Hierarchy
k	k = Create symbolic linK
K	SHIFT K = create Knotpoint B-spl.
l	l = Create Level object
l	CTRL l = Create Line light source
p	p = Create Point light source
v	v = Create camera from View

Frame Buffer

F	SHIFT F = Open external screen = Framebuffer
Q	SHIFT Q = Close external screen = Quit framebuffer

Grids

^c	CTRL c = Create Grid
^d	CTRL d = Delete Grid
^g	CTRL g = Grid Snap On/Off
^m	CTRL m = Modify grid
^r	CTRL r = Reposition grid
^s	CTRL s = Select grid
^v	CTRL v = Grid Visible on/off

Macros

x	x = EXecute current macro
---	---------------------------

y	y = Repeat current macro
z	z = Execute named macro
M	SHIFT M = Record Macro On/Off

Modify Hierarchy

°	ALT	b = Cut
ç	ALT	c = Copy
ð	ALT	d = Duplicate
©	ALT	e = Swap = Exchange selected
¶	ALT	p = Paste selected
\$\times\$	ALT	x = delete = eXterminate selected

Modify Properties

a	a = Modify Animation
c	c = Modify Color
i	i = Modify attribute Information
n	n = Modify Name
t	t = Modify Tags

Modify Shape

d	d = Deform object
e	e = ShEar object
j	j = Move cog = Jump
m	m = Move object
o	o = MirrOr object = Opposite position
p	p = Project to Object
q	q = Rotate and extend = looks like a Q
r	r = Rotate object
s	s = Stretch object
1	1 = Size 1d = extend
2	2 = Size 2d
3	3 = Size 3d
I	SHIFT I = Inverse kinematic
j	CTRL J = Bend circular in 2D
k	CTRL K = Twist

Modify Freeform

C	SHIFT C = Freeform surface to Curves
D	SHIFT D = Freeform Distribute
E	SHIFT E = Freeform Exchange u & v
G	SHIFT G = Freeform assiGn
J	SHIFT J = Freeform concatenate = Join
O	SHIFT O = Freeform Open/close
P	SHIFT P = Freeform reParametrize
R	SHIFT R = Freeform Remap

S	SHIFT S = Freeform Swap directions
T	SHIFT T = Freeform modify Type
V	SHIFT V = Freeform inVert
X	SHIFT X = Freeform Delete Point or Isoparam. Curve
Y	SHIFT Y = Freeform Insert Point or Isoparam. Curve
Z	SHIFT Z = Freeform Break Curve or Mesh

Object Loading/Saving

i	ALT	i = Insert object
®	ALT	r = Replace object
ß	ALT	s = Save object

Rendering Boxes

^a	CTRL a	= Delete All boxes
^b	CTRL b	= Define Box
^e	CTRL e	= DElete box = Erase
^w	CTRL w	= shoW all boxes

View Settings

+	+ = Zoom in
-	- = Zoom out
*	* = Toggle parallel/perspective projection
/	/ = Toggle accurate/bounding box wire
.	. = Display Position
^f	CTRL f = Auto Focus
^h	CTRL h = Render Hierarchy
^i	CTRL i = Separate Io on/off
^x	CTRL x = Define X axis
^y	CTRL y = Define Y axis
^z	CTRL z = Custom Zoom scale
^o	CTRL o = Object space to view
^n	CTRL n = No gadgets

Vector Stack

L	SHIFT L = Lasso selector
---	--------------------------

Windows

F1	F1 = View window to front, if not open open it
F2	F2 = Select window to front
F3	F3 = Tool window to front
F4	F4 = Material window to front
F5	F5 = Palette window to front
F6	F6 = Screen window to front

F7 F7 = Animation window to front
 F8 F8 = RPL Shell window to front
 F9 F9 = Measuring System to front
 F10 F10= Close active window

F10 SHIFT F1 = Close View windows
 F11 SHIFT F2 = Close Select windows
 F12 SHIFT F3 = Close Tool windows
 F13 SHIFT F4 = Close Material windows
 F14 SHIFT F5 = Close Palette windows
 F15 SHIFT F6 = Close Screen windows
 F16 SHIFT F7 = Close Animation windows
 F17 SHIFT F8 = Close RPL Shell windows
 F18 SHIFT F9 = Close Measuring window
 F19 SHIFT F10= Close active window

Miscellaneous

ALT n = Project New
 ALT q = Quit program

1.3 keybindings2

DEFAULT KEYBINDINGS - by key

ALSO SEE by category

KEYPAD

* toggle parallel/perspective projection
 / toggle accurate/bounding box wire
 + Zoom in
 - Zoom out
 . Display Position
 (unused
) unused
 0 unused
 1 extend 1Dimensions modify/linear/extend
 2 size2D 2Dimensions modify/linear/size2D
 3 size3D 3Dimensions modify/linear/size3D
 4 unused
 5 unused
 6 unused
 7 unused
 8 unused
 9 unused

NORMAL KEYS 'a'...'z'

a change animation modify/properties/animation
 b step backwards animate/control/step_backwards

c	change colour	modify/properties/colour
d	deform object	modify/linear/deform
e	shEar object	modify/linear/shear
f	step forward	animate/control/step_forwards
g	create group	create/structure/group
h	create methOd level	create/structure/method
i	Info about attributes	modify/properties/attributes
j	move cog = Jump	modify/linear/move COG
k	create linK	create/structure/link
l	create Level object	create/structure/level
m	move	modify/linear/move
n	change name	modify/properties/name
o	Opposite position	modify/linear/mirrOr
p	create point lightsource	create/light/point
q	rotate and extend	modify/linear/rot&ext
r	rotate	modify/linear/rotate
s	stretch	modify/linear/strech
t	change tags	modify/properties/tags
u	Update anim. system	animate/control/refresh
v	Create camera from View	view/camera/create
w	unthink and rethink Wireframe one level	
x	eXecute current macro	project/macros/execute current
y	repeat current macro	project/macros/repeat current
z	execute named macro	project/macros/execute named

SHIFT KEYS 'A'...'Z'

A	boolean And with paint	create/boolean/And with paint
B	Boolean or	create/boolean/or
C	surface to Curves	modify/Freeform/surf.to curves
D	Distribute points	modify/freeform/distribute
E	Exchange u & v	modify/freeform/exchange_u&v
F	open Framebuffer	project/external screen/open
G	freeform assiGn	modify/freeform/assign
H	edit anim. methOd	animate/edit
I	Inv. kinematics	modify/special/inv.kinem
J	Join freeforms	modify/freeform/concatenate
K	create Knotpoint B-spl.	create/controls/B-Spline_Knot
L	Lasso selector	extras/vectors/lasso select
M	record Macro on/off	project/macros/record
N	boolean AndNot with paint	create/boolean/and not with paint
O	freeform Open/close	modify/freeform/open close
P	freeform reParametrize	modify/freeform/reparametrize
Q	Quit framebuffer	project/external screen/close
R	freeform Remap	modify/freeform/remap
S	freeform Swap directions	modify/freeform/swap direction
T	Type of freeform	modify/freeform/type
U	unthink all Wireframe	create/boolean/unthink
V	freeform inVert	modify/freeform/invert
W	unthink and rethink Wireframe in all sublevels	
X	Delete point/curve	modify/freeform/delete
[go to beginning	animate/control/go_to_beginning
]	go to end	animate/control/go_to_end
>	play forward	animate/control/play_forward
<	play backwards	animate/control/play_backwards

&	boolean and	create/boolean/and
%	boolean and not	create/boolean/and_not
.	display Position	view/camera/display_pos
?	go to ?	animate/control/go_to_?

FUNCTION KEYS `F1`...`F10`

F0 project Window to front, if not open it
 F1 select
 F2 tool
 F3 Material
 F4 Color
 F5 Screen
 F6 Animation
 F7 Shell
 F8 Measuring
 F9 close active window

SHIFT FUNCTION KEYS `F1`...`F10`

F10 view view window close
 F11 select window close
 F12 tool window close
 F13 material window close
 F14 color window close
 F15 screen window close
 F16 animation window close
 F17 rpl shell window close
 F18 measuring window close
 F19 close active screen

CTRL KEYS: Codes 0...31, `^A`=1...`^Z`=26

CTRL A	delete All boxes	view/boxes/delete all
CTRL B	define Box	view/boxes/define
CTRL C	Create grid	view/grid/create
CTRL D	Delete grid	view/grid/delete
CTRL E	dElete box = Erase	view/boxes/delete
CTRL F	auto Focus	view/viewcam/autofocus
CTRL G	Grid snap on/off	view/grid/snap to grid
CTRL H	render Hierarchy	view/render/render hierarchy
CTRL I	separate Io on/off	view/type/separate io
CTRL J	Bend circular in 2D	modify/bend circular/move 2d
CTRL K	Twist	modify/non-linear/twist
CTRL L	Create Lightline	create/light-sources/line
CTRL M	Modify grid	view/grid/modify
CTRL N	No gadgets	project/window/no gadgets
CTRL O	Object space to view	view/input plane/object->iplane
CTRL P	Reset Hot-Point	view/input plane/reset hot point
CTRL Q	General Settings	settings/general
CTRL R	Reposition grid	view/grid/reposition
CTRL S	Select grid	view/grid/select
CTRL T	change objtype	modify/freeform/type
CTRL U	clear Undobuffer	settings/undo/clear

```

CTRL V    grid Visible on/off  view/grid/visible
CTRL W    shoW all boxes      view/boxes/show all
CTRL X    define X axis       view/viewcam/define x
CTRL Y    define Y axis       view/viewcam/define y
CTRL Z    custom Zoom scale   view/viewcam/set custom

```

ALT KEYS `a`...`z`

```

ALT A      unused
ALT B      cut = delete & Backup selected  modify/structure/cut
ALT C      Copy selected                    modify/structure/copy
ALT D      Duplicate selected                modify/structure/duplicate
ALT E      swap = Exchange selected          modify/structure/swap
ALT F      can't be used!
ALT G      can't be used!
ALT H      can't be used!
ALT I      Insert object                     project/objects/insert
ALT J      can't be used!
ALT K      can't be used!
ALT L      unused
ALT M      unused
ALT N      project New                       project/project/new
ALT P      Paste selected                    modify/structure/paste
ALT Q      Quit program                      project/exit real
ALT R      Replace object                    project/objects/replace
ALT S      Save object                       project/objects/save
ALT T      unused
ALT U      unused
ALT V      unused
ALT W      unused
ALT X      delete = eXterminate selected     modify/structure/delete
ALT Y      unused
ALT Z      unused

```

1.4 tags

TAGS

Type Characters:

Type	Explanation
C	- Control tag, used only internally.
F	- Floating-point tag.
I	- Integer tag.
S	- String tag
V	- Vector tag
M	- Modifiable vector tag. Treated as an absolute 3D point.
D	- Modifiable vector tag. Treated as a vector.

Reserved Tag Indentifiers

ID	Explanation
CEND	- This ends the tag data structure.
DDIR	- Primary direction vector for object.
DDIV	- Secondary direction vector for object.
ISKE	- Some animation methods like PATH DIRECTION and SKELETON add attributes (VPHS and VDIR/VDIV tags) to their target objects when the animation system is refreshed for the first time. This tag is used for indicating that all required data is defined.
MCOG	- Center of Gravity.
SCRE	- Formula producing logical result to control target creation by CREATION method. The result should be assigned to the 'l' variable.
SDEL	- Logical formula to control target deletion by CREATION method. The result should be assigned to the 'l' variable.
SMAT	- This tag is used for defining materials associated with mapping objects. The tag value contains the name of the material.
SMTH	- The name of the method associated with method objects.
SOBJ	- Reference to another object. Links and groups refer to other objects using this tag.
SRPL	- The contents of this tag can be any RPL program. When associated with method or parameter object, the tag can be used for customizing methods and evaluable parameter objects.
VFRQ	- Defines how much faster (or slower) a method's time runs compared to its parent time.
VOFF	- Offset vector used by several methods.
VPHS	- General usage phase tag. The tag is used for modifying a method's local time. When associated with target objects, the purpose of the tag depends on the method in question.
VTIE	- Time end tag. When the time reaches this value, the method stops.
VTIM	- Method's current time.
VTIS	- When the current time reaches this value, the method is activated.
FFRI	- Surface friction between particles involved in Collisions.
FMAS	- Mass for object.
FREB	- Rebound energy for Collision detection system. The default value is 1.0 (fully elastic); the value 0 results to fully non-elastic behaviour.

FSIZ	- Size for particle. This tag can be used for overriding the default object size.
ICSM	- Accuracy for Collision detection system. Possible values 0 ... 2.
SFOR	- String tag used for defining evaluable formulas.
VSPI	- Spin for particles.
VVEL	- Velocity of particles.
SIDE	- Unique identifier used for linking group and link primitives with their targets.
SWND	- Window name. Can be added to viewpoint and aimpoint.
IFLG	- Method evaluation specifier
ITRA	- Force type of a force method
FORC	- The strength of the force
IIND	- Inv. Kinematic joint specifier
IOCT	- Octaves in fractal noise method
ILMP	- Light source intensity distribution
ILBR	- Light source brightness
FLRD	- The radius for the local fading of light sources
FLSF	- The light "spot/beam edge smoothing" percentage.

1.5 materials

REAL 3D MATERIAL EDITOR

Common Material Properties
Material Handlers
Material Variables
Procedural Materials

1.6 material properties

COMMON MATERIAL PROPERTIES

Field	Description
Name	Name of material in the Material library.

Texture	The path and name of an image file to use for defining various material properties like color or transparency.
Spline	B-spline meshes itself are used for the mapping definition.
S-map uvwh	The position and size of the image file when it is mapped onto a spline. The u and v values determine the position of the top left corner of the image and w and h control what proportion of the spline is covered. Each of these can be between 0.0 and 1.0.
Color map	Texture is to be used for material color definition.
Bump map	Red component of texture is used for bump map evaluation. The brighter the red component, the higher the bump.
Transparency map	Green component is used for transparency evaluation. The brighter the green component, the more transparent the material.
Brilliance map	Blue component is used for evaluating brilliance. The brighter the blue component, the more mirrorlike the material.
Shadow map	The RGB values of each pixel of the texture file are used to modify the current color values for the corresponding point of the objects surface.
Clip map	The surface of the object is clipped by the texture file. The object surface is removed wherever it is not covered by the texture file. This includes any areas not covered by the mapping or tiling, or any areas selected as transparent.
Scope mask	The material effect is modified by using the texture as a mask, which defines where the material is applied. Only the points which are affected by the texture, included in the material, get non-zero scope. The application test is equivalent to the one used for clip mapping.
Transparent Color (Transp. col)	The color to use as the transparent color. This affects the application of an texture for clip mapping and scope mask. Values vary between 0 and 255.
Unshaded	Light sources and shadows do not affect the shading of this material.
Smooth	Removes specular reflections on the boundaries of transparent materials.
Exclusive	Causes other material definitions to be ignored for any points on the surfaces covered by that specific material.

Tile	Selects whether texture file is to be tiled in X, Y or both directions.
Flip	If set, then every second tile is flipped, making texture map edges match better.
X-Freq. & Y-Freq.	Specify the number of tiling repetitions over the surface of the texture.
Grade	Selects whether color gradients are calculated for X, Y or both directions.
Specularity	Controls how sharply defined are the high-lights reflected from the surface of the material by light sources. The higher the specularity, the smaller the high-light and the harder its edges.
Specular brightness (Spec. bright.)	This affects how intense the specular high-lights are.
Brilliance	The degree to which light is reflected directly from the material surface ('mirror-like' property).
Transparency	The degree to which light passes through the material surface.
Turbidity	Controls the degree to which light is affected as it passes through the material. The higher the setting the denser the 'fog'.
Turbidity saturation (Turbid. sat.)	Defines how the distance which light travels in a turbid material affects the light. The default value 25 gives a linear result; so if the distance is doubled, then the effect is doubled. If this value is zero, then the distance has no effect at all.
Refraction	This determines the degree to which light is bent as it passes through the material. It represents the speed of light in the material as a percentage of its maximum speed through empty space. The higher this value the less the light is bent.
Roughness	This controls the degree of 'molecular texture' applied by the material. This 'molecular' texture is a random bump-map which is independent of the magnification of the material.
Dither	This enables dithering of material color to be applied to individual objects and the precise amount of dithering to be selected.
Bump height	Relative scale of bumps produced by bump-mapping texture file and procedural bump handler.
Effect	Controls how strongly the properties of the current

material will affect the objects to which it is applied.

1.7 material handlers

MATERIAL HANDLERS

Mapping
Scope
Bump
Color
Index

1.8 mapping

MAPPING

VARIABLES

x, y, z .

DESCRIPTION

Using an equation to modify one of these variables changes the way in which a texture file is mapped onto the surface.

BUILT-IN PROCEDURES

Tilt $y = y + x*a*h/w$
Waves $y = y + \sin(x*a*PI/w)*b*h$
SwapXY $tmp = x, x = y, y = tmp$
Noise Fractal noise with amplitude b and density factor a

1.9 scope

SCOPE

VARIABLES

$s, sp, sb, br, tr, tu, ts, ro, ri$

DESCRIPTION

The primary variable for the scope handler is 's' which, along with Effect level, determines how much of the material properties are mixed with the material properties already applied to the objects.

BUILT-IN PROCEDURES

```
Sphere      if r < a s=100, else s=0
InvLin      s = 100/(1 + r^2/a)
InvExp      s = 100*exp(-r/a)
Local       s = max(100*a/(a-r),0)
Temporal    s = s*(a*(1 - T) + b*T)
Noise       Fractal noise with amplitude b and density factor a
```

Where a = sz by default, except in Temporal, where a = 1 by default.

ALSO SEE

Material Variables

1.10 bump

BUMP

VARIABLES

bh, bx, by.

DESCRIPTION

The variables bx & by define the vectors used to deviate the normal and produce the bump-mapping effect. If an texture file is being used as a bump map, then bx and by are first evaluated from the red component of the texture.

BUILT-IN PROCEDURES

```
Waves  bx = bx + sin(x*a*PI/w)*b
Bumps  bx = bx + sin(x*a*PI/w)*b, by = by + sin(y*a*PI/h)*b
Noise  Fractal noise with amplitude b and density factor a
```

1.11 color

COLOR

VARIABLES

R, G, B.

DESCRIPTION

The initial values for R,G & B are evaluated from the texture file if used. Mathematical formulas can then be used to

modify or replace these initial values, as with the other Material Variables. The size variable (sz) can be used for to bind a formula to the size of a texture.

BUILT-IN PROCEDURES

```
Bright  R = R*b/(r*a + 1.0) ( G,B similar )
Waves   R = R + sin(x*a*w/h)*b ( G,B similar )
Granite  Intensity = Intensity*Fractal noise with amplitude b
         and density factor a
Noise    R = Intensity*Fractal noise, G and B similarly
Marble   Intensity = Intensity * 'Zone-magnified' Fractal noise
```

ALSO SEE

Material Variables

1.12 indexh

INDEX

VARIABLES

i

DESCRIPTION

The material variable i is evaluated by any index format string used in the texture file name. Using a mathematical formula based upon either T, t or Frm makes it possible to control the indexes of texture files in very flexible ways to create moving material textures.

BUILT-IN PROCEDURES

```
Default  i = a, a+1, ..., b-1, 0, 1, ..., b-1, ...
PingPong i = a, a+1, ..., b-1, b-1, b-2, ..., 0, 1, ...
```

Index handler default values are a = 0.0, b = 0.0 (zero offset and no modulo cycle).

ALSO SEE

Material Variables

1.13 material variables

MATERIAL VARIABLES

Variable	Type	Description
----------	------	-------------

a, b	Float	User definable variables, initially assigned the value of the numeric gadgets to the right of the expression gadget.
x	Float	Horizontal texture coordinate.
y	Float	Vertical texture coordinate.
z	Float	Depth texture coordinate.
sz	Float	Size of texture geometry.
r	Float	Distance from the origin of the texture.
s	Float	Scope output variable.
sp	Float	Specularity
sb	Float	Secular brightness
br	Float	Brilliance
tr	Float	Transparency
tu	Float	Turbidity
ts	Float	Turbid saturation
ri	Float	Refraction index
ro	Float	Roughness
di	Float	Dithering scale
bh	Float	Bump height
bx	Float	Bump-map horizontal coefficient
by	Float	Bump-map vertical coefficient
R	Integer	Red color component
G	Integer	Green color component
B	Integer	Blue color component
t	Float	Local Animation time
i	Integer	Material texture index
Frm	Integer	Current Frame index
Res	Integer	Frame Resolution
T	Float	Global Animation time

The ranges of x & y are either between 0.0 and 1.0 if no texture mapping type is being used, or between zero and the number of pixels along the x or y dimension of the texture file.

z, sz & r are distances expressed in spatial coordinates and can have any positive value.

If the mapping used is type Default, then x,y & z are the absolute spatial coordinates, and any mathematical handlers will effectively use Parallel mapping along z-axis.

The value sz depends on the texture geometry in the following way:

- parallel - length of the shorter texture rectangle edge
- cylinder - radius of the cylinder (average if elliptic)
- sphere - radius of the sphere (average if elliptic)
- disk - radius of the disk (average if elliptic)

The values for R,G & B are from 0 to 255. If a user defined formula assigns a value greater than 255, then it will be limited to 255. Negative values become zero.

The range for T & t is between 0.0 and 1.0.

The value of i is assigned by the user and can have any integer value.

Frm & Res are a positive integer values between 0 and MAX_INT.

All the others should be between 0.0 and 100.0. If a formula takes a variable outside this range, then the effects are unpredictable.

The order of the handlers indicates the order in which they are evaluated; Mapping first and Index last.

Although the material variables can be assigned values at any time, assigning them values before the handler in which they are properly assessed has no practical effect, e.g. The color components are assigned their values from the texture file after the evaluation of the mapping and scope.

The material variables x,y & z can be modified in any of the expressions.

1.14 procmat

PROCEDURAL MATERIALS

Note that this information can't be found in the manual!

The Real 3D Material window contains five Material Handler cycle gadgets. Each gadget contains the option RPL which can be used for associating procedures with the material in question. These procedures can define all material properties, such as brilliancy, color, bumps etc. by fly through Material Variables which are defined during rendering.

In order to create a procedural material, the following steps are required:

1. An RPL procedure must be written and saved to a file.
2. Desired 'Handler' gadget in the Material Window must be set to RPL and the corresponding 'Expression' gadget must contain the procedure call.
3. The name of the procedure file must be associated with the material by using the menu Define/Procedures of Material window.

The following things should be kept in mind when designing procedures:

1. Use variables 'a' and 'b' for passing required data to the procedures. If your procedure requires more than two parameters, use the normal RPL parameter passing mechanism for extra parameters.
2. Never parametrize the size or direction of the procedure, because that can be achieved simply by modifying the size of the mapping primitive.
3. The default value for a & b variables in the material window is 0.0. Design your procedures to produce reasonable results with these default values.
4. Do not create procedures which can fail (for example, to division by zero error). However, if your procedure simply can't carry out its job for whatever reason, use the ERROR word to terminate the rendering with the proper error message.
5. The procedure file associated with a material is executed every time the rendering engine is started. If more than one material refers to the same file, it is executed once per each material. If the file contains RPL words or variables, they must be enclosed with ?IF ?ENDIF words in order to prevent redefinitions.

The file 'procmat.rpl' found in the directory 'R3D2:rpl' contains some example procedures. For more information, consult that file. The name of the procedure reflects the Method field it is associated with. For example, if the name of the procedure is BumpWaterDrop, it should be used with the 'Bump' method.

1.15 animation system

ANIMATION SYSTEM

- Built-in Animation Methods
- Custom Animation Methods
- Evaluable Objects
- RPL Variables

1.16 built-in methods

BUILT-IN METHODS

PATH	ROTATION
SWEEP	SIZE
STRETCH	DIRECTION
MOVE & DIR	CONTROL CURVES
SIMPLE SKELETON	SKELETON
INV KINEMATIC	MORPHING OPEN
MORPHING CLOSED	TRANSFORM
WAVE	RADIAL FORCE
DIRECTED FORCE	TANGENT FORCE
COLLISION	INT COLLISION
FRICTION	CREATION
PROCESSOR	RPL
NOISE	ATTRIBUTES

1.17 path

PATH

SYNTAX

```
      Level
     /    \
Targets Level (M)
         \
         path
```

PARAMETERS

path - Evaluable object

TAGS

VPHS, ISKE

VARIABLES

a, b, c - relative movement during 'dt'
dt - time interval
t, u, v - current time

1.18 rotation

ROTATION

SYNTAX

```

      Level
     /    \
Targets Level (M)
         \
         axis

```

PARAMETER

axis - Any primitive

VARIABLES

i, j, k - spin
l - Modify flags (0/4/8)

1.19 sweep

SWEEP

SYNTAX

```

      Level
     /    \
Targets Level (M)
         /    \
      center ctrlcurve

```

PARAMETERS

center, ctrlcurve - Evaluable objects

TAGS

VPHS - Phase used for defining rotations for the target

VARIABLES

l - Modify Flags

1.20 size

SIZE

SYNTAX

```

      Level
     /    \

```



```

Targets  Level (M)
        /      \
        center ctrlcurve

```

PARAMETERS

center, ctrlcurve - Evaluable objects

VARIABLES

l - Modify flags

1.21 stretch

STRETCH

SYNTAX

```

          Level
         /    \
Targets  Level (M)
         /      \
        coordsys ctrlcurve

```

PARAMETERS

coordsys - coordsys primitive
ctrlcurve - any evaluable object

VARIABLES

l - Modify flags, see ROTATE method.

1.22 direction

DIRECTION

SYNTAX

```

          Level
         /    \
Targets  Level (M)
         \
         path

```

TAGS

VPHS, ISKE

1.23 move & dir

MOVE & DIR

SYNTAX

```
      Level
     /   \
  Targets Level (M)
         /   \
      motion direction
```

TAGS

VPHS, ISKE

PARAMETERS

motion, direction - Evaluable objects

1.24 control curves

CONTROL CURVES

SYNTAX

```
      Level
     /   \
  Targets Level (M)
         /   \
      curve1 curve2
```

PARAMETERS

curve1, curve2 - evaluable parameters

TAGS

VPHS, ISKE

VARIABLES

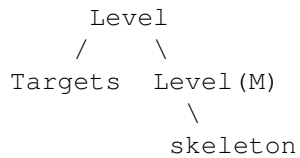
1 - Modify flags. The first bit is used to define whether or not the method should stretch the target. If set, targets are not

stretched.

1.25 simple skeleton

SIMPLE SKELETON

SYNTAX



PARAMETERS

skeleton - Evaluable object

TAGS

VOFF - The offset between the skeleton and target COGs.

VPHS - Parameter value defining the position on the skeleton.

SFOR, SRPL - when associated with targets, can be used for redefining the position on the skeleton.

ISKE - the value 2 indicates that the VPHS and MCOG tags defined for each target.

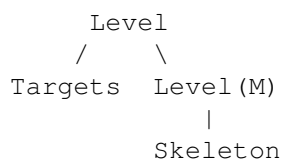
VARIABLES

x, y, z - Parameter space defining the position of the target object on the skeleton.

1.26 skeleton

SKELETON

SYNTAX



PARAMETERS

Skeleton - Evaluable object

TAGS

VOFF - displacement between target COG and skeleton objects

VPHS - parameter value defining the position on the skeleton

SFOR, SRPL - when these tags are associated with the targets, variables x, y and z can be used for redefining the position of the target object.

ISKE - Indicates that the VPHS, VDIR, and VDIV tag definitions are executed.

VARIABLES

x, y, z - position of the target object.

1.27 inv kinematic

INV KINEMATIC

SYNTAX

```

      Level
     /    \
Skeletons Level (M)
           \
           endpoint
  
```

PARAMETERS

endpoint - evaluable parameter that defines the end point for the skeletal object

TAGS

VOFF - Offset vector.

SRPL, SFOR - Formula/procedure for defining the end point

VPHS - Phase for end point evaluation

IIND - the joint to be modified: IIND=0 = default end point, IIND = 1 modifies the previous point etc.
This tag is added to the method.

VARIABLES

a, b, c - endpoint

1.28 morphing open

MORPHING OPEN

SYNTAX

```

      Level
     /    \
  Target  Level (M)
       / | \
    key1 key2 ...

```

PARAMETERS

key1, key2, ... - two or more key-frame objects

TAGS

VOFF - displacement for the target object
 FKNO - Time value (0 - 1) of a key object
 IMIT - Interpolation type (this is a method tag)
 0 = Linear, 1 = B-Spline

1.29 morphing closed

MORPHING CLOSED

SYNTAX

```

      Level
     /    \
  Target  Level (M)
       / | \
    key1 key2 ...

```

PARAMETERS

key1, key2, ... - two or more key-frame objects

TAGS

VOFF - displacement for the target object

1.30 transform

TRANSFORM

SYNTAX

```

      Level (M)
      /      \
    axis    Curve

```

PARAMETERS

coord - axis or coordsys primitive to which the evaluated point from 'trans' is projected.

trans - time is mapped to the parameter space of this evaluable parameter.

1.31 wave

WAVE

SYNTAX

```

      Level
      /      \
    Targets Level (M)
           /      \
        coordsys  curve

```

PARAMETERS

coordsys - coordsys or axis primitive defining wave direction

wave - any evaluable object defining a shape of the wave

TAGS

None

VARIABLES

x, y, z - position of the target in the object space of 'coordsys'

fx, fy, fz - position corresponding x, y and z variables

t, u, v - current time

1.32 radial force

RADIAL FORCE

SYNTAX

```
      Level
     /   \
Targets Level (M)
         \
         center
```

PARAMETERS

center - evaluable primitive defining a 'center' of the force.

TAGS

FMAS - mass
VVEL - velocity

VARIABLES

x, y, z - COG of target object
a, b, c - velocity of target object
i, j, k - spin of target object
d - size (diameter) of target
m1 - mass of target object
m2 - mass of parameter object
dt - time interval between subsequent animation samples
e - kinetic energy of target
f - strength of the force
t, u, v - current time
fx, fy, fz - direction of the force (unit vector)
s - distance between parameter target objects

1.33 directed force

DIRECTED FORCE

SYNTAX

```
      Level
     /   \
Targets Level (M)
         \
         direction
```

PARAMETERS

direct - evaluable parameter defining the direction and center of the force field

TAGS

FMAS - mass (kg)
 VVEL - velocity (m/s)
 VSPI - spin (rad/s)

VARIABLES

x, y, z - COG
 a, b, c - Velocity
 i, j, k - Spin
 d - size (diameter)
 m1 - mass
 dt - duration
 e - kinetic energy
 f - strength of the force
 t, u, v - current time
 fx, fy, fz - direction of the force field (unit vector)
 s - distance between center of force field and the object
 in question

1.34 tangent force

TANGENT FORCE

SYNTAX

```

      Level
     /   \
Targets Level (M)
         \
         axis
  
```

PARAMETERS

axis - evaluable parameter defining the axis of the rotating
 cylindrical field of force.

TAGS

FMAS - mass
 VVEL - velocity
 VSPI - spin

VARIABLES

x, y, z - COG
 a, b, c - velocity
 i, j, k - spin
 d - size
 m1 - mass
 dt - duration

e - kinetic energy
 f - strength of the force
 t, u, v - current time
 fx, fy, fz - direction of the force field
 s - distance between center of the force and the object in question

1.35 collision

COLLISION

SYNTAX

```

      Level
     /    \
Targets Level(M)
      / | \
    obj1 obj2 ...
  
```

PARAMETERS

obj1, ... - objects with which the target objects can collide.

TAGS

FREB - Rebound Energy (0 ... 1)
 FFRI - Surface Friction (0 ...)
 ICSM - Collision Surface Sampling (0, 1, 2)
 FMAS - Mass
 VSPI - Spin
 VVEL - Velocity
 FSIZ - Size of the bounding sphere

VARIABLES

m1 - mass of the collided parameter object
 m2 - mass of the collided target object
 s - distance between COGs
 a, b, c - relative velocity vector ($v_2 - v_1$)
 i, j, k - relative spin
 e - relative kinetic energy
 t, u, v - current time
 o1, o2 - addresses of collided objects
 l - 1 = process collision, 2 = do not process collision, 3 = fatal error
 p1, p2 - addresses of internal collision data structures

1.36 int collision

INT COLLISION

SYNTAX

```
      Level
    /      \
Targets  Level (M)
```

TAGS

FREB - Rebound Energy (0 ... 1)
FFRI - Surface Friction (0 ...)
ICSM - Collision Surface Sampling (0, 1, 2)
FMAS - Mass
VSPI - Spin
VVEL - Velocity
FSIZ - Size of the bounding sphere

VARIABLES

m1 - mass of the collided parameter object
m2 - mass of the collided target object
s - distance between COGs
a, b, c - relative velocity vector (v2 - v1)
i, j, k - relative spin
e - relative kinetic energy
t, u, v - current time
o1, o2 - addresses of collided objects
l - 1 = process collision, 2 = do not process collision,
3 = fatal error
p1, p2 - addresses of internal collision data structures

1.37 friction

FRICTION

SYNTAX

```
      Level
    /      \
Targets  Level (M)
```

TAGS

FMAS - mass
FSIZ - size
VVEL - velocity
VSPI - spin

VARIABLES

x, y, z - COG (position) of the object
 a, b, c - velocity
 i, j, k - spin
 d - size (diameter)
 m1 - mass
 dt - time interval (duration)
 e - kinetic energy
 f - coefficient of friction
 t, u, v - current time

1.38 creation

CREATION

SYNTAX

```

      Level
     /    \
Targets  Level (M)
        /  |  \
      sample1 ....
  
```

PARAMETERS

sample - sample objects for procedural creation

TAGS

VCRE - creation time
 SCRE - formula used for procedural creation
 SDEL - formula used for procedural deletion

VARIABLES

x, y, z - position of the object
 a, b, c - velocity
 i, j, k - spin
 d - size
 m1 - mass
 dt - time interval
 e - kinetic energy
 t, u, v - current time
 fx, fy, fz - birth day
 l - boolean value for deletion/creation

1.39 processor

PROCESSOR

SYNTAX

```

      Level
     /   \
Targets Level (M)

```

VARIABLES

```

a, b, c - velocity
i, j, k - spin

```

1.40 rpl

RPL

SYNTAX

```

Level (M)
 \
  ?

```

PARAMETERS

The number and type of parameters required for this method are entirely dependent upon the implementation of the method procedure attached to it via the SRPL tag.

TAGS

SRPL - RPL procedure to be executed

VARIABLES

Any

1.41 noise

NOISE

SYNTAX

```

      Object
     /   \
Targets Noise (M)
          \

```

CoordSys

PARAMETERS

CoordSys - The size and direction of this parameter
define the density distribution of the noise field.
The smaller the parameter, the denser the fractal noise

TAGS

VVEL, VSPI - Maximal Velocity/Spin change by the noise
IOCT - Octaves in the noise
IFLG - Spin/Velocity modify selector

The abovementioned tags are attached to the method level.

1.42 attributes

ATTRIBUTES

SYNTAX

```

      Object
      /    \
Targets  ATTRIBUTES (M)
          /    \
        Sample1 Sample2

```

PARAMETERS

Sample1 - The attributes of this object are copied to target
objects when the time is between the start and end time
of the method.
Sample2 - The attributes of this object are copied to target
objects when the time is outside the method time line.

1.43 custom methods

CUSTOM METHODS

These methods are defined in the file 'methods.rpl' and are completely
implemented using RPL.

All methods defined in the file can be installed by adding the following
line to the 's:rpl-startup' file:

```
"methods.rpl" LOAD
```

or executing the file as a macro.

```
ABS PATH
CHAIN
WEIRD FORCE
```

1.44 abs path

ABS PATH - absolute motion for target objects

SYNTAX

```
      Level
     /   \
Targets Level (M)
         \
         path
```

TAGS

No

VARIABLES

No

DESCRIPTION

Moves the COGs of target objects along a given path. If any of the target objects are dislocated from the path for whatever reason, it is immediately pulled back to the curve.

1.45 chain

CHAIN

SYNTAX

```
      Level
     /   \
Targets Level (M)
```

TAGS

FDIS - the distance between subsequent targets

DESCRIPTION

Attempts to keep the distance between subsequent targets equal. The tag "FDIS" can be associated with the method object in order to define the distance between targets. If the tag is not defined, the default distance 0.5 is used.

1.46 weird force

WEIRD FORCE - particle system oriented method example

SYNTAX

```
      Level
     /    \
Targets  Level (M)
```

DESCRIPTION

The WEIRD_FORCE demonstrates how to create physical oriented 'particle-system' methods. The method generates random force field affecting to the velocity and the spin of target objects.

1.47 evaluable objects

EVALUABLE OBJECTS

The following primitives can be used as evaluable parameters:

- Offset
- Axis
- Coordsys
- Ellipse
- Line
- Mesh
- Skeleton

Other primitives can be made evaluable by attaching SFOR or SRPL Tags to them and by defining relevant RPL variables.

The following variables can be modified by user defined formula/procedure:

- x, y, z - position
- i, j, k - direction

1.48 rpl variables

ANIMATION SYSTEM ORIENTED RPL VARIABLES

Variable	Description
T	- The current time
Res	- Frame resolution
Frm	- Current frame
a, b, c	- Velocity
i, j, k	- Spin
x, y, z	- Center of Gravity
t, u, v	- Local time of the method in question or parameter value for object evaluation
m1, m2	- Mass
d	- Diameter of the object (size of the bounding sphere)
f	- Strength of the Force
rnd	- Random value, always between 0 and 1
o	- Address of the object
l	- General usage 32 bit integer value
fx, fy, fz	- General usage variables. Purpose depends on the context
s	- Distance
dt	- Time interval between subsequent animation samples
e	- Kinetic energy of the object

1.49 rendering

RENDERING SETTINGS

Output

Selects output target for rendered image:

File

When one of the 'File' output targets is selected, then the name of the destination file is entered here.

Mode = Draft

The rendering engine uses a grey-scale evaluation of the object color and ignores all material properties to render the image.

Mode = Environment
All objects are treated as Not Reflected with reflections being taken from Environment color and/or map. Only a single light-source from the view-point is used.

Mode = Lampless
The scene is rendered using full object and material properties, but only the single view-point light-source is used.

Mode = Shadowless
All user-defined light-sources are evaluated, but no shadows are calculated.

Mode = Normal
Full rendering evaluation.

Mode = Outline
The scene is rendered as a hidden-line wire-frame image by rendering the edges of all objects.

Dithering = Rnd RGB
Separate random deviation for each color component:

Dithering = Rnd intensity
The same random deviation is used for each component:

Dithering = Fixed rnd int
The same random deviation for each color component and a fixed dithering pattern is used for every frame.

Dithering = Row
Colors dithered line by line.

Dithering = Raster
Uses a checkered pattern for dithering.

Dithering = None
No dithering applied.

Ambient
Color and level of ambient light.

Background
Color of image background. This does not interact with the rendering of objects and materials.

Environment
This specifies the color which is evaluated as if an infinite sphere of this color surrounds the objects in the scene.

Brightness
This controls the scaling of all the light sources in the scene.

Overlight
The level of this setting controls how rapidly the color intensity turns the color to pure white.

Recursions

This defines to what depth light rays are evaluated as they reflect from surface to surface.

Dither scale

This defines the maximal deviation of the color signals when using dithering.

Backdrop image

When the Backdrop image gadget is enabled, then the named file is used as a background to the rendered scene.

Environment map

Setting the Environment map gadget maps the file specified onto the 'environment sphere'.

Width & Height

These gadgets control the width & height of the rendered image in pixels when rendering to a file or an External Screen.

Pixel h/w

Controls the aspect ratio used for individual pixels when rendering.

DOF Scale

Depth of Field scale. The higher the value, the more rapidly distance from the Aim-point increases blurring.

DOF Strength

This numeric controls how much blurring occurs at a given distance.

X/Y-resolution

These two gadgets control the size of patches evaluated when rendering.

Antialiasing

This controls when the color signal difference triggers the Adaptive Over-sampling of the Anti-aliasing routines.

Lightsamples

The amount of sampling used for diffuse light-sources.

Mat. samples

Amount of sampling for Non-homogeneous Material Properties.

Subdivisions

Controls how finely B-spline surfaces are evaluated when rendering, and if B-spline->Phong is set, how much each face is subdivided.

B-spline->Phong

B-spline surfaces are converted internally to phong type freeforms before rendering.

Autoexp

The effect of this is the same as that of automatic exposure by a camera, which is to produce the most balanced image possible

under the available lighting conditions.

Field rendering

Every odd frame is rendered half a pixel lower.

No bgr. antial.

Prevents anti-aliasing between the edges of objects and the background.

Alpha output

Rendering calculations will be carried out using Alpha Information from visibles.

HL-shading

Uses additive instead of proportional method to calculate consecutive shades of a color.
