3 Configuration

Preferences

Viewport

OpenGL Shading, p.54

*New option: OpenGL Lighting*

If this option is enabled, OpenGL will be used to light your scene instead of CINEMA 4D’s software shading. This will usually result in faster display in the viewport. OpenGL lighting supports the following features:

**Environment Object**
- Environment color
- Fog (3D view only)

**Light**
- Light types: Omni, Spot, Distance, Parallel
- Color and brightness
- The No Diffuse and No Specular options

**Materials**
- Luminance
- Specular, incl. Width and Height
- Specular Color
- Texture in the Color channel with Specular

**Restrictions:**
- Some light parameters are not supported.
- Only the first eight light sources are used.

✔ You can test the performance of your graphics card using CINEBENCH 2003, the free benchmarking tool that is based on CINEMA 4D. CINEBENCH 2003 is included with the R8.1 update.
Import/Export

Shockwave 3D Export, p.66

Bones
You can now export any number of bones. However, only the nine most influential bones will be used per point. The reference position of the bones will determine which bones have the most influence.

Lights
A maximum of eight lights can be used for the current view. Additional lights will be ignored.

VRML1 Import, VRML2 Import, Wavefront Import, p.75
These three import filters now have an additional option: Import Normals. So what is this new option for?

Some NURBS modeling applications tend to export mesh that is prone to software shading errors. With the VRML 1, VRML 2 and Wavefront formats, you can now include vertex normals in the file (provided that you have activated vertex normals in the modeling application). Vertex normals are normals for points.

Depending on how many surfaces are involved, a point may even have several normals. With the help of these vertex normals, CINEMA 4D is then able to produce good results despite the sub-optimal mesh. The vertex normals are stores in a Normal tag, which is displayed in the Object manager:

These Normal tags enable the imported objects to be displayed and rendered correctly. Keep in mind that if you edit the points of these imported objects, the Normal tags will no longer work properly and the objects will be shaded incorrectly. However, you can move, rotate and scale the object as a whole without ruining its shading. If you delete the Normal tag, the usual shading will be used instead.

6 Edit Menu

The drawing pipeline

Use Expressions, p.120

✔ You can also switch off XPresso expressions by clicking this icon.
7 Objects Menu

Lighting

Sun Light, p.297

Contrary to the description in the reference manual, north is represented by the positive Z-axis, south by the negative Z-axis, east by the positive X-axis and west by the negative X-axis.

Managing your environment

Foreground Object, Background Object, p.383

You can now scale and move background pictures interactively, by adjusting the Offset and Length settings in the Attribute manager (Texture tag selected).

Selection Object, p.386

When you create a Selection object, the objects that are currently selected will be assigned to it. This gives you a new, third way to assign objects to a Selection object.
10 Structure Menu

Edge Melt
Using this new tool, you can ‘melt’ the shared edge of two neighboring triangles. This will convert the triangles into a single quadrangle. If this quadrangle is non-planar, shading anomalies may appear in the viewport. However, the anomalies will disappear when you render (the render engine treats each non-planar quadrangle as two triangles).

New Phong Shading commands

Break Phong Shading
Phong shading is broken for the selected edges (see ‘Phong Tag’, later in this addendum). The Phong tag’s Angle Limit option will be enabled automatically if it isn’t already.

Unbreak Phong Shading
Phong shading is restored for the selected edges. If you are not working in Edge mode, Phong shading will be restored for all broken Phong edges.

Select Broken Phong Edges
Selects all broken Phong edges.

Structure Context Menu

Snap Settings, p.493
You can now switch snapping on or off without exiting the Spline mode. To do this, first assign a shortcut for snapping in the Command manager. Then when you are working in Spline mode, press the shortcut to toggle snapping on or off.

12 Plugins

FlashEx, p.514
FlashEx has been optimized. It now offers faster export and uses less memory. In addition, FlashEx is better able to cope with high numbers of polygons and defective polygons.
If you still encounter problems, try the following:
- Remove defective polygons (optimize the objects).
- Use as few polygons as possible (the Polygon Reduction tool may help here).
- If you are using HyperNURBS objects, reduce the number of subdivisions (FlashEx uses the Subdivision Renderer value).

14 Window Menu

Console, p.585

The much-missed Reload C.O.F.F.E.E. Plugins command has been returned to the Plugins menu. The following restrictions apply:
- Does not work with “Message” and filter plugins.
- Affects existing plugins only (new plugins cannot be loaded).
- Registration values are not updated, such as Name, ID and Icon.
17 Object Manager

File Menu

Phong tag, p.608

New Phong Shading algorithm

CINEMA 4D’s Phong shading has been improved to resolve several shading issues. This especially applied to mesh imported from NURBS modeling applications such as Rhino 3D. Problem areas such as groups of long, thin triangles are now rendered smoothly.

In addition, edges can now be taken into account with a new option, Use Edge Breaks. See the illustration above. You can now create edge breaks in CINEMA 4D using the new command Break Phong Shading (Structure menu).
When breaking individual edges, the effect will only occur if the end points of the edge are surrounded by connected polygons. In the following illustration, the effect of breaking will be visible for edge A but not for edge B.

➔ Broken Phong edges only work when the Phong tag’s Angle Limit option is enabled. If you don’t want to use an angle, set Angle Limit to 180 (this corresponds to no angle limit).

You can enable or disable the effect via the Use Edge Breaks option.

The following types of NURBS object make use of these new Phong features: Extrude, Lathe and Sweep. To see the difference, select the NURBS object’s Phong tag and in the Attribute manager, disable the Use Edge Breaks option. Provided the option is enabled, the edges will be sharp.

**Protection Tag, p.610**

➔ The Protection tag prevents scaling with the Object tool, but not with the Model tool.

**Stick Texture Tag, p.611**

*Limitation*

If you assign a Stick Texture tag to a texture that uses Flat mapping with Side set to Front, the ‘Front’ information will be lost and the texture will be mapped to both sides of the object. This is not a bug but a limitation of Stick Texture tags.
18 Material Manager

The Material Editor

Environment, p.658

The Environment page contains a new option: Exclusive. If this option is enabled, environment reflections will only appear in those places where there are no ‘real’ reflections (Reflection channel on).

Illumination, p.670

On this page you’ll find a new setting: Preview Size. It controls the resolution of textures in the viewport. You can choose values from 64x64 to 1024x1024. The higher this value, the better the quality of the texture in the viewport, but also the more memory it needs. Keep in mind that high resolution textures can slow down OpenGL.

→ The Default setting corresponds to 256x256.
19 Timeline

Animation Toolbar

Recording keys, p.833

The menu that appears when you click this icon has two new options. They will only be available if a Selection object is selected from the menu.

Restrict Editor Selection

If this option is enabled, the only objects you will be able to select in the viewport are objects that are assigned to the Selection object.

Restrict Keyframe Recording

Applies to recording with the Record button only. Keys will be recorded for all objects that are assigned to the Selection object, regardless of which objects are currently selected.

To record keys with the Record command, p.833

Selection objects will also be taken into account when you record keys manually with the Record button.

Edit Menu

Automatic Mode, p.876

In the Timeline, a small triangle is displayed next to the name of each object. Click a triangle to reveal its object’s tags.

➔ XPresso Expression tags also have triangles; click these to reveal the nodes that belong to the expressions.
20 F-Curve Manager

F-Curves in the Timeline, p.891

If you set a key’s interpolation to Custom, its tangents are displayed.
- Shift-drag to move the left or right tangent arm independently of the other arm.
- Alt-drag to change the length only of the tangent (the direction will be locked).

21 Attribute Manager

The Attribute manager’s context menu

Add Keyframe Selection, p.905

For example, if you define the color of a light as a keyframe selection, the subchannels are highlighted in red, not the light itself. The same applies to all parameters that are composed of subchannels.

Other enhancements
- In common with the Timeline, The F-Curve manager now has a time slider.
- If you Shift-drag a time slider (be it in the animation toolbar, the Timeline or the F-Curve manager), the scene won’t animate. This enables you to record the current setup at a different frame.

22 XPresso Editor

XGroups, p.916

Active

This option activates or disables the XGroup. Nodes within the XGroup will only be calculated if the XGroup is active.
Protect / Unprotect

This button enables you to protect XGroups by means of a password. The contents will then be hidden. Only by entering the correct password can the user gain access to the contents of the XGroup.

To protect an XGroup, first lock its contents (from the context menu, enable View > Locked). In the Attribute manager, click the Protect button. In the dialog that opens, enter a password for the XGroup. The XGroup is now protected.

The procedure for unprotecting an XGroup is almost the same in reverse. First click the Unprotect button then enter the password into the dialog that opens. To make the contents of the XGroup visible once more, unlock the XGroup (disable View > Locked from the context menu) and choose the desired display mode (context menu: View). For more information on locking and unlocking XGroups, see Chapter 22, ‘XPresso Editor’, in the CINEMA 4D reference manual.

Choosing the data type, p.920

Filename

This new data type is used for storing a file path. For example, the Sound node uses the Filename data — for storing the path of the sound file that you want to load.

XPresso Nodes

Object

Relative Reference, p.936

The example path given in the reference manual is incorrect. It should read as follows:

For example, the path ‘UPPDN’ stands for ‘Up, Previous, Previous, Down, Next’, where Previous and Next represent a jump to the previous or next element on the same hierarchical level.

The same also applies to ‘Path’ on page 937.

Start Position, p.936

This Object

Contrary to the entry in the reference manual, this mode is not identical to the Absolute Reference mode (if you assign the XPresso tag to a different object in the Object manager, this object will be referenced instead).
Object node, p.937

Inputs and outputs

Various input and output ports are available to each object node. The types of these ports will depend on the type of object or tag involved. These ports are too numerous to be documented here in full. However, you can usually tell what a port does by its name in the XPresso editor. For example, the Global Position ports give you the position of the object in 3D space.

Ports on the left side of the node are input ports — they pass data to the object, material or tag. Ports on the right side are output ports and pass the data on to one or more other nodes.

One Object node port worth a special mention is the On port. It is available to all Object nodes but not to Material or Tag nodes. The On port enables or disables the node via a value of the type Bool; this is useful for controlling when the node’s data should be passed and when it should be halted.

The node is enabled if its On port receives a Bool value of True (1) or disabled if it receives a Bool value of False (0). The node will be enabled automatically if you do not add the On port.

Point, p.938

Example

Here the node outputs two values: the object’s total number of points (here, eight) and the position of the point whose index value is 4.

Not only can the Point node output the position of a specific point, it can also output the point’s normal, as a vector. The vector is exactly one unit in length and is always perpendicular to the point. The direction is calculated using the positions of the surrounding surfaces. Hence if the object has no surfaces, the point normals cannot be calculated accurately.

Random, p.939

Contrary to the description in the reference manual, the Random node outputs decimal numbers in the range of -1.0 to +1.0. Two modes are available for generating these numbers: Free and Time. Choose the desired mode in the Attribute manager.

The node can output random values of the following data types: Bool, Integer, Real or Vector. Specify the desired data type via the node’s output port.

In addition, the Positive Only option, in the Attribute manager, enables you to specify whether the random values will be negative and positive (option disabled) or positive only (option enabled). However, if the node outputs random values of the type Bool, the option will not affect the result (a Bool value can have a value of 1 or 0 only).
Result, Spy, p.939

Result and Spy nodes slow down the expression. Therefore you should delete them as soon as you have finished using them.

Time, p.940

You can access various animation parameters via the following output ports:

**Time**
Current time is seconds since the start of the animation (internally, this is more precise than ‘Real’).

**Real**
Current time in seconds since the start of the animation.

**Frame**
Current frame since the start of the animation.

**Frames per second**
The frame rate as defined in the Project Settings.

**Start**
The start of the animation as defined in the Project Settings.

**End**
The end of the animation as defined in the Project Settings.
Loop start
The start of the preview area (the green area in the Timeline) in seconds.

Loop end
End of the preview area in seconds.

Delta
The time interval between two frames in seconds.

Previous
The time before the current frame.

Iteration, p.941
This node has been moved to the new Iterator group. In addition, a new input port and a new output port have been added.

The On input port takes a Bool value that enables (True) or disables (False) the node. The node is enabled automatically if you do not add this port.

The Count port outputs the total number of iterations in the loop, i.e. Iteration End - Iteration Start.

➔ You can connect several Iteration nodes together via their Previous Iterator ports to form nested loops.

The example in the reference manual has a wire missing, which is marked above. The position of each point in Cube 1 is assigned to the position of each corresponding point in Cube 2.

What would happen without this critical wire?
Cube 2’s points would all be set to the same position as the first point of Cube 1 (point number 0 in the Structure manager).
Polygon, p.942

Example

This node outputs two values: the object’s total number of polygons (in this case, six) and the position of the fifth polygon’s center (index value 4).

Invert, p.947

You can use this node to invert numbers or matrices. For example, you can combine this node with a MatrixMulVector node to convert a position from world coordinates to local coordinates. In the Attribute manager, choose whether a number or a matrix should be inverted.

Cross Product, p.947

The order of the input vectors affects the direction of the output vector. If you swap the input vectors around, the output vector will point in the opposite direction.

The direction of the output vector depends on the direction of the plane between the two input vectors, but not on its length. Hence you may need to normalize the output vector (for this, use a Universal adapter node and set its Data Type to Normal). In other words, you may need to set the length of the output vector to 1.

Range Mapper, p.950

In the reference manual, the description for ‘Input Range, Output Range’ is incomplete. Here is the missing information:

Input Range, Output Range

Here you can choose commonly-used ranges such as 0% to 100%. The following ranges are available:

User Defined

In this mode, the input range and the output range are defined by the values in the Attribute manager for Input Upper, Input Lower, Output Upper and Output Lower (or by the ports of the same name, if you have added these).

Degree

Defines the range 0 to 360.
*Radians*
Defines the range 0 to 6.283. This corresponds to 2*PI.

*Percent*
Defines the range 0% to 100%.

*Zero To …*
These modes define the ranges 0 to 1, 0 to 100 and 0 to 1,000.

**Spline Resolution**
This new option enables you to define the accuracy of the spline, from Normal to Best. In general, use a high resolution when there is a big difference in size between the input range and the output range, or when the spline frequently changes direction over a small part of the graph.

**FloatMath, p.953**
The description in the reference manual is incorrect. The difference between the Math node and the FloatMath node is as follows:

**Math Node**
This node carries out its calculations using the data type defined in the Attribute manager. If the node receives input values of a different data type, they are converted to the Math node’s data type before the calculation takes place.

**FloatMath Node**
This node, on the other hand, always carries out its calculations using the Real data type. If the node receives input values of a data type other than Real, they will be converted to the type Real before the operation is performed. In the case of an input value of the type Vector, the length of the vector will be used.

**Compare, p.954**
The following function is now available.

* =
Not Equal To. The output is True if the two input values are not equal. Otherwise the output if False. This function is, therefore, the reverse of the Equality operator (== ).
Equal, p.955

The Equal node has a new option: Not Equal.

Not Equal

This option, which is available in the Attribute manager, reverses the action of the node. With the option enabled, the output is True if the two inputs are not equal. Otherwise the output is False.

Set Driven Keys

Relative Driven Keys, p.958

Contrary to the description in the reference manual, the commands Set Driven (Absolute) and Set Driven (Relative) both create a Range Mapper node. In both cases, the values for Input Upper/Lower and Output Upper/Lower are adjusted accordingly.

New Nodes

Bounding Box

The maximum dimensions of an object are displayed in the viewport as red corners. These corners form a ‘bounding box’, inside which the object fits. The dimensions of this bounding box are displayed in the Coordinate manager when the Size drop-down list is set to Size.

The Bounding Box node gives you access to the dimensions of this bounding box. The object must be a polygon object. If you want to use the node with a primitive object, you must first make the primitive editable.

In addition, if Used Deformed Points is enabled, deformed points will be taken into account to ensure an accurate result. The Matrix Mode setting enables you to choose local coordinates or global coordinates.

This example queries the size of the bounding box for a spline object. This Bounding Box node also outputs the smallest and largest coordinates of the corner points that make up the bounding box.
After connecting the Object port to the desired object, you can access the following information via the output ports:

**Box Minimum**
Outputs the coordinates of the smallest corner point of the bounding box, as a vector.

**Box Maximum**
Outputs the coordinates of the largest corner point of the bounding box, as a vector.

**Box Size**
Outputs the maximum dimensions of the bounding box in the X, Y and Z directions, as a vector. Keep in mind that, in contrast to the Coordinate manager, the choice of coordinate system has a major influence on the result.

**Box Point 1 to Box Point 8**
Outputs the coordinates of the corner points that make up the bounding box.

**Collision**

This node is able to detect collisions. Connect the two objects that you want to check for collision to the Object 1 and Object 2 input ports. These objects must be polygon objects. If you want to use the node with a primitive object, you must first make the primitive editable.

In the Attribute manager, choose the collision detection mode:

**Bounding Box**
In this mode, the node checks whether the bounding boxes of the two objects intersect each other. If you want child objects to be taken into account as well, in the Attribute manager enable Include Children. The two bounding boxes will be expanded if necessary to enclose the child objects as well.

**Sphere**
A virtual sphere is placed around each of the objects. If these two virtual spheres intersect, the node detects a collision. If Include Children is enabled in the Attribute manager, these two virtual spheres will also enclose the child objects.

**Object**
This mode detects collisions between the two objects at the polygon level. This is the most accurate type of collision detection but also the most CPU-intensive. In this mode you can also apply materials to the surfaces that are colliding. To do this, create a Polygon Selection tag each for the two objects then drag and drop these tags into the Selection Tag boxes in the Attribute manager. The node will now store the polygons that are colliding in these Selection tags.
Distance

Detects a collision when the surfaces of the two objects are within a certain distance of each other. This distance is defined by the Tolerance input port.

Tolerance

In this mode, the Tolerance input port defines a distance around each object. This distance increases the virtual size of the objects for the purposes of collision detection. If, taking their virtual sizes into account, the objects then intersect each other, a collision is detected.

In Object mode, colliding polygons can be assigned different materials. Create a Polygon Selection tag each for the two objects then drag these tags into the node’s Selection Tag boxes. Assign a material to each object and restrict these materials to the appropriate Polygon Selection tag.

If the objects colliding are deformed, enable Use Deformed Points to ensure accurate collision detection.

The Matrix Mode settings in the Attribute manager are only relevant when the node outputs the point positions of the objects. The following output ports are available:

Collision

Outputs a value of the type Bool which indicates whether the objects are currently colliding.

Distance

Outputs the smallest distance between the objects (works in the Sphere and Distance modes only).

Point 1

If a collision has been detected, this port outputs the position vector of the point in Object 1 that collided first (provided Collision Type is set to Object). The exact value of the position depends on the coordinate system chosen and whether the object is being deformed (see the options in the Attribute manager).
Point 2

If a collision has been detected, this port outputs the position vector of the point in Object 2 that collided first (provided Collision Type is set to Object). The exact value of the position depends on the coordinate system chosen and whether the object is being deformed (see the options in the Attribute manager).

Polygon Index 1

If a collision has been detected, this port outputs the index number of the polygon in Object 1 that collided first (provided Collision Type is set to Object or Distance).

Polygon Index 2

If a collision has been detected, this port outputs the index number of the polygon in Object 2 that collided first (provided Collision Type is set to Object or Distance).

FlipFlop

This node is a Boolean switch. Three input ports are available. If the On input port is enabled, the node outputs the value 1. If the Off input port is enabled, the node outputs the value 0 (even if the On port is still enabled). The same applies in reverse: if the Off port is enabled first followed by the On port, the result is 1.

If the Switch port is enabled, the result is reversed. A value of 1 becomes a value of 0, and vice versa. If Switch is enabled and then the value of one of the other ports changes, the result is once again affected — the result always depends on the port that was last enabled. For example, if the On port was the last port enabled, the result will always be 1; if the last port enabled was the Off port, the result will always be 0. Note that disabling a port has no influence of the result. It is only the enabling of these ports that causes the node to change its output.

Link List

The Link List enables you to introduce a limited number of objects into the expression. Drag the desired objects from the Object manager and drop them into the Link List box in the Attribute manager. You can control which object in the list the node outputs via the Index input port. If you want the node to output all the objects in the list, use an Iteration node.
The following input port is available:

**Index**

This port enables you to access a particular object from the list. For example, pass a value of 0 to this port to access the first object in the list, a value of 1 to access the second object in the list, and so on. If the Index value is greater than the number of objects in the list, the index loops back to the start of the list again. So if, for example, there are two objects in the list and the Index value is 2, the node will output the first object (0 = first object, 1 = second object, 2 = first object again, and so on).

This expression uses the number of objects in the list to control an Iteration node. The Iteration node is connected to a copy of the Link List node, which outputs all the objects one after another.

The following output ports are available:

**Count**

Outputs the total number of objects in the Link List.

**Link**

Outputs the index number of the object selected in the Link List.

**MonoFlop**

The MonoFlop node is a Boolean switch with a built-in time delay. Once triggered, the node counts down a specified number of frames. The node outputs a Bool value of True while it is counting down. In the Attribute manager, choose the desired countdown mode:

**Normal**

The countdown beings once the Trigger port’s value changes from 1 to 0. In other words, nothing happens when the node is enabled. Only once the node is disabled will the clock start ticking.

**OneShot**

The node starts counting down as soon as the value 1 arrives at the Trigger port.

For both modes, the following applies: If the node is triggered while in the middle of a countdown, the countdown starts again from the beginning.
The following input ports are available:

**Duration**

This controls the duration of the countdown. If you set this value in the Attribute manager, enter the duration in frames. However, if you connect, say, a Constant node to this port, the value defines the duration time in seconds.

**Reset**

If this port receives a Bool value of 1, the node’s internal counter is reset to 0.

**Time**

This port enables you to, for example, connect a Time node to the MonoFlop. The Time node can then be used to control the speed of the countdown.

**Trigger**

This port takes a Bool value and is used for starting the countdown.

The MonoFlop node in this example is enabled every nine frames. The MonoFlop node then remains active for 0.2 seconds before becoming inactive again.

The following output ports are available:

**Count**

Outputs the total number of times that the node has been triggered. You can reset this counter by passing a value of 0 to the Reset input port.

**Out**

Outputs a Bool value of 1 if a countdown is in progress. Otherwise the port outputs a value of 0.

**State**

If the node is currently counting down, this port outputs a decimal value between 0.0 and 1.0 that indicates how far the countdown has progressed. For example, if the countdown is 30% of the way through the countdown, this port will output the value 0.3. If on the other hand the node is disabled, the port outputs the value 0.
Noise

This node uses noise to generate random numbers. Four types of noise are available: Noise, Turbulence, Wavy Turbulence and Fractal Brown Movement.

Turbulence is composed of several frequency bands of summed noise. Wavy Turbulence is the same as Turbulence, except that the changes are more abrupt (‘wavefronts’). Lastly, Fractal Brown Noise is the same as Turbulence, except that its high frequency bands are weaker.

Noise values depend not only on the animation time, but on coordinates as well. This enables the node to output different values for the same frame of the animation. In this example, each point is given a different value.

The following input ports are available:

**Amplitude**

This value scales the result of the node by multiplying it. An Amplitude of 1 leads to noise results in the range of -1.0 to +1.0.

**Frequency**

The Frequency controls the generation of the noise structure. Higher frequencies lead to more changes in noise per unit of time.

**Octaves**

Defines the level of detail for the noise structure. Increase this value for more detail and greater fluctuation of the values.

**Scale**

This value scales the noise structure (not the result!).
Seed

This value is a starting point for the calculation of the noise structure. The seed value will therefore lead to repeatable results. If you are using several noise nodes with identical settings, they will each generate the same results unless you give them different Seed values.

Time

The noise structure can be animated. By default, the animation’s current time is used. This will cause a new noise value to be calculated for each frame of the animation. However, you can connect your own time values to this port (such as the time values created by some of the Thinking Particles nodes).

Vector

You can pass a vector to this input port. This will enable a three-dimensional noise structure to be calculated.

X Coordinate, Y Coordinate, Z Coordinate

To control the generation of three-dimensional noise fields, you can, for example, connect the coordinates of object points to these input ports. Each point can then be assigned a different value that is dependent of the point’s position in 3D space.

ObjectIndex

Some nodes are able to access objects, materials or tags indirectly, such as by their position in the Object manager or Material manager. The ObjectIndex node enables you to check whether the element found using one of these nodes really is an object, tag or material. The node will only pass through these permitted elements and thus can prevent errors from happening later on in the expression.

The Object Index node outputs the objects it receives from the Hierarchy node. This enables the expression to detect collisions for multiple objects.

Connect the element that you want to check to the Instance input port. Provided that the element meets the criteria, the node will output it via the Instance output port. The node will also output the total number of other elements of the same type that are on the same hierarchy level, via the Index port. The ObjectIndex node is frequently used in combination with the Hierarchy node.
Ray Collision

This node generates a ray and checks if and where it hits a polygon object. If you want the ray to collide with primitive a object or a NURBS object, you must convert the object to polygons before using it with this node.

The following input ports are available:

Object
Here, connect the polygon object that you want to fire the ray at.

Ray Point1, Ray Point2
The ray is defined by two points. The vector between these two points represents the ray. Ray Point1 defines the starting position of the ray, Ray Point2 its end position. These two points enable you to define not only the direction of the ray, but its length also.

The ray has been created between points P1 and P2. The expression places the sphere at the point where the ray hits the cube.

The following output ports are available:

Collision
This port signals whether the ray has hit the object. A Bool value of 1 indicates a collision, a Bool value of 0 means the ray missed the object.
**Distance**

If the ray has hit a polygon, this port outputs the distance from Ray Point1 to the polygon first hit by the ray. To switch on this port, in the Attribute manager, disable Test Only.

**Face Index**

If a collision has taken place, this port outputs the index number of the first polygon hit by the ray. To switch on this port, in the Attribute manager, disable Test Only.

**Face Normal**

If the ray has hit a surface, this port outputs the normal of the polygon hit. To switch on this port, in the Attribute manager, disable Only Test.

**Hit Position**

Outputs the position where the ray first hits the object (recall that the ray travels from Ray Point 1 to Ray Point2). This vector can only be calculated if a collision has taken place. To switch on this port, in the Attribute manager, disable Test Only. The Hit Position is given in either local or global coordinates, depending on whether the Global Coordinates option is enabled in the Attribute manager.

**Is Backface**

This port outputs a Bool value which indicates whether the ray hit a backface (output is 1) or a frontface (output is 0). Note that the direction of the polygon’s surface normal determines which side is the front and which is the back.

**Phong Normal**

In contrast to Face Normal, this port’s output depends on the normals of the polygons that surround the polygon hit by the ray. To switch on this port, in the Attribute manager, disable Only Test.

**Reference**

With this node, you can access an object using a relative search path. This path is described by the letters ‘U’, ‘D’, ‘N’ and ‘P’. Use ‘U’ and ‘D’ to move up or down a hierarchy level respectively, and ‘N’ and ‘P’ to move to the next or previous element respectively. The Object node also references objects in this way — see the description of the Object node for more details.
This Reference node finds the object which is directly below the Cube object (Path = ‘D’ for ‘Down’), i.e. the Sphere object. Once found, the object (i.e. the Sphere) is then passed to an Object node, which in turn outputs the object’s name: ‘Sphere’.

The following input ports are available:

**Instance**

The object that you connect here defines where the search path begins. For example, to reference the children of a particular object, connect the object to this port. The path then starts from this object.

**Path**

Connect the string that contains the path to this port. The path itself is composed of the characters already mentioned: ‘U’, ‘D’, ‘N’ and ‘P’. You can also enter this path directly into the Attribute manager. The node will output the object described by the path via the Instance output port (provided that an object can be found at that position).

**Sound**

With this node, you can load and play back sound files in the viewport (but not in the render). The sound file must be in a format that CINEMA 4D supports, such as WAV or AIFF. To load the file, in the Attribute manager, click the button that is next to the Filename text box. Use the dialog that opens to locate and open the sound file.

In the Attribute manager, you’ll also find a setting called Probe. This defines how many samples will be used to calculate the amplitude — the greater the number of samples in this calculation, the more damped the sound will become. In other words, the sound gets quieter as you increase the Probe value.

The following input ports are available:

**Play**

Pass a Bool value of 1 to this port to trigger sound playback.
Sample

A sample is the smallest unit that a sound file can have. The number of sound samples per second can vary greatly depending on the compression and quality of the sound file. This port gives you access to every sample.

Time

Usually you’ll want the sound to be played back at its original speed. However, if you want to play the sound at a faster or slower rate, connect your own time to this port.

A sound is played each time the cube and sphere collide.

The following output ports are available:

Left

Outputs the current volume of the left sound channel (this is the amplitude of the samples for the left sound channel).

Length

Outputs the length of the sound file in seconds. If you want the number of samples instead, use the Sample port.

Right

The current volume of the right sound channel (this is the amplitude of the samples for the right sound channel).

Samples

The total number of samples in the sound file.

Spline

This node outputs the coordinates of positions along a spline. You specify the position that you want to check as a percentage, where 0% represents the start of the spline and 100% represents the end of the spline. Depending on the Matrix Mode setting in the Attribute manager, the node outputs the position in local coordinates or in global coordinates.
A spline can consist of one or more segments. If the spline has more than one segment, pass the number of the segment that you want to examine to the Segment input port. Note that the first segment is accessed with the value 0, the second segment is accessed with the value 1, and so on.

If Use All Segments is enabled in the Attribute manager, the Offset value refers to the length of the entire spline including all its segments. If the spline is composed of one segment only, you can ignore the Segment port (the segment will be used automatically).

The Iteration node outputs all whole numbers from 0 to 100 one after another. These numbers are each divided by 100 to convert them to Real values in the range 0.0 to 1.0. These Real values are then passed to the Spline node's Offset port, which causes the Spline node to output the coordinates of 100 positions along the spline.

The way in which a spline is subdivided depends on its interpolation settings. For example, points may be concentrated around the bends. This has implications if you want to move an object along the spline at constant velocity — the spline must be subdivided evenly. Enable the Use Natural Distribution option if you want the Spline node to output values that are equal distances apart.

The following output ports are available:

**Closed**
Outputs a Bool value of 0 if the spline in open or 1 if the spline is closed (splines can be closed by enabling the Close Spline option in the Attribute manager).

**Length**
Outputs the length of the chosen spline segment, as a Real number.

**Position**
Outputs the coordinates of a position along the spline (as defined by the Offset value and the Matrix Mode setting).

**Segments**
Outputs the total number of spline segments. When passing the number of a segment to the Segment input port, keep in mind that the number of the first segment is 0.

**Tangent**
Tangents define the run of the curve before and after the position along the curve. This port outputs the tangent for the chosen position along the spline in Vector format.
**Weightmap**

Weight maps store a value between 0.0 to 1.0 for each point in an object. These values affect how strongly each point is influenced when deformers such as bones are applied to the object. With the Weightmap node, not only can you read these weight values, you can change them also.

The following input ports are available:

**Index**

The values in the weight map are stored in the same order as the index values of the object’s points. Therefore the first weight value in the list applies to point 0, the next weight value to point 1, and so on. To access the weight value of a specific point, pass the point’s index number to this port. If you are not sure how many points the object has or how many entries there are in the weight map, you can find this out using a Point node.

An object’s first Vertex Map tag is accessed. The first point (index value = 0) is then assigned a weight of 0.5 (i.e. 50%).

**Tag**

Connect the desired tag to this port.

**Value**

Here you can set the point’s weight. You can enter a value from 0.0 to 1.0.

The following output ports are available:

**Count**

Outputs the object’s total number of points

**Value**

Outputs the point’s weight value.
**Colorscape**

Colors in CINEMA 4D can be defined using any of several different colorspaces. This node enables you to convert a color value from one colorspace to another. Choose the type of conversion in the Attribute manager.

One use of the Colorscape node is to find out how bright an RGB color is. Convert the RGB value to HSV; the V value gives you the color’s brightness.

If RGB From 0 To 255 is enabled, the components of RGB colors will be in the range of 0 to 255 instead of the normal range of 0.0 to 1.0. If Hue 0.0 To 360.0 is enabled, hue values are given in degrees instead of radians.

**Dot Product**

The dot product, or scalar product as it is often called, calculates the projection of a vector onto another vector. You can think of this as the shadow that is cast onto a sun dial, where one of the vectors lies on the dial plate, the other vector represents the direction of the gnomon (the bit that sticks out and casts the shadow) and the sun is shining down from directly above the sun dial. This shadow corresponds to the dot product of the two input vectors.

Here the dot product is used to calculate the angle between two input vectors. The following applies: the dot product of Vector 1 and Vector 2 = the length of Vector 1 * the length of Vector 2 * the cosine between Vector 1 and Vector 2.
You can connect the two input vectors either way round to the Input 1 and Input 2 ports (the result will be the same no matter which way round you connect the input vectors). The inputs must always be vectors, but you can change the data type of the node itself in the Attribute manager. You have a choice of Color, Vector or Normal.

**MatrixMulVector**

A local vector is a vector whose components have been calculated using an object's local coordinate system. If you multiply a local vector by the object's global matrix, the result is a global vector. Therefore among other things you can use this node to convert a point's local position to world coordinates. You can also perform the reverse operation (i.e. convert world coordinates to local coordinates) by first inverting the object's global matrix (use an Invert node — set the Invert node's Data Type to Matrix).

This expression converts one of the cube's coordinates from local to global coordinates.

**C.O.F.F.E.E.**

This node enables you to integrate C.O.F.F.E.E. code with your XPresso expression. You can add any number of input ports to the node and rename them via the node's context menu. However, you cannot change the type of a port. If you add a port of the wrong type, you must delete it (double-click it) then add a new port of the correct type.

The input ports are declared automatically as variables — there is no need to declare them in the node. For example, the default C.O.F.F.E.E. node has two input ports (Input 1 and Input 2) which are used in the code without first being declared. As with the input ports, you can add output ports of varying type and you can rename them freely.
You can enter the expression directly into the Script box in the Attribute manager, or for more space you may prefer to use the C.O.F.F.E.E. editor instead. To open the C.O.F.F.E.E. editor, in the Attribute manager, click the Open C.O.F.F.E.E. Editor button. From this editor, you can also compile the program and run it.

The Protect button enables you to protect your expression by means of a password. The contents will then be hidden. Only by entering the correct password can the user gain access to the contents of the XGroup. If you’re the type of person who doesn’t like remembering passwords, why not just keep your own unprotected copy of the expression?

Keep in mind that there are some restrictions when using C.O.F.F.E.E. programs in C.O.F.F.E.E. nodes. In particular, neither the project nor the objects in the scene can be accessed directly by the C.O.F.F.E.E. code. Instead pass the desired information — such as an object’s position — to the C.O.F.F.E.E. node via another node.

The C.O.F.F.E.E. node is designed primarily for evaluating math expressions and for adding simple programming structures to the XPresso expression, such as the if/then/else statement. Complex expressions should be written as standard C.O.F.F.E.E. expressions instead.

**Iterator (Group)**

These nodes have one thing in common: they can each output many values per frame. For example, a single iterator node can output the position of each object in the scene. Some of these nodes were previously available in different groups. Now you will find all such nodes here, on the Iterator menu.

**Hierarchy**

This node first locates a position in an object hierarchy, as defined by a Reference Path and a Start Path. From this position in the hierarchy, the node then follows a repeatable Iteration Path and outputs all of the objects it meets as it travels along this iteration path. The node is controlled primarily in the Attribute manager.
Reference Mode

Use Start Position

To reference an element relative to the object that owns the XPresso Expression tag, use either this mode or the more advanced Relative Reference mode. For example, to reference the element that is three positions before the object that owns the tag, set Reference Mode to Use Start Position, set Start Position to Predecessor On This Level and set Start Distance to 3. The Use First Position mode also enables you to reference the first or last object in the scene.

Absolute Reference

This mode references the object that is displayed in the Reference box. If you want to use the expression in several places in your scene, avoid using the Absolute Reference mode, otherwise you’ll need to change the referencing of the objects each time.

Relative Reference

This mode is similar to Use Start Position. It enables you to reference an element relative to the object that owns the XPresso Expression tag. Unlike the Use Start Position mode, you enter a path that gives the node instructions on how to reach the element. The starting position is the object that owns the tag. For example, the path ‘UPPDN’ stands for ‘Up, Previous, Previous, Down, Next’, where Previous and Next represent a jump to the previous or next element on the same hierarchical level. Up and Down represent jumps one level up or one level down the hierarchy. The advantage of a relative reference is that it enables you to transfer the Object node from one hierarchy to another without problems. Edit the path using the Path box.

Start Position

This defines the Start Position when Reference Mode is set to Use Start Position. The setting is relative to the object that owns the XPresso Expression tag. For example, if you set Start Position to First In This Level, the node references the first element on the same hierarchical level as the object that owns the tag.

This Object

The object that owns the tag.

Up In Hierarchy

Chooses the first object one level up the hierarchy.

Down In Hierarchy

Jumps to the first object one level down the hierarchy.

Successor On This Level

Chooses the next object on the same hierarchical level.

Predecessor On This Level

Jumps to the previous object on the same hierarchical level.
First On This Level

Chooses the first element on the same hierarchical level.

Last On This Level

References the last element on the same hierarchical level.

First In Scene

Chooses the first object in the scene.

Last In Scene

Jumps to the scene’s last element.

Start Distance

With some Start Position modes you might want to jump several steps in the hierarchy in one go. In such cases, use Start Distance as a multiplier for the Start Position. For example, to reference an object that is three positions after the object that owns the XPresso Expression tag, set Start Position to Successor On This Level and set Start Distance to 3.

Reference Path

See ‘Relative Reference’, above.

Start Path

This box is available only when Reference Mode is set to Relative Reference. Use the box to enter the path for the relative reference. For example, ‘UPPDN’ stands for ‘Up, Previous, Previous, Down, Next’. (See also ‘Relative Reference’ above.)

Iteration Path

The Hierarchy iterator continually repeats this path. For example, ‘N’ will cause the node to output all objects that are on the same level one after another.

The Hierarchy iterator continues to output objects until either of the following happens:

- there are no more objects at the hierarchical position defined by the Start Path and the (repeated) Iteration Path.
- the number of iterations defined by the Maximum Iterations input port has been reached.

The Hierarchy node then refers to the object that is defined by the Reference Path and Start Path once more. The next time the node is called, the Iteration Path will be repeated over again.

Exclude

If you want the node to ignore some of the objects in the hierarchy during the iteration, drag the names of these objects from the Object manager and drop them into this box. The iteration will skip over any of these objects it meets as it runs through the hierarchy.
Paths enable you to access groups of objects at any position in the hierarchy. In this example, the Hierarchy node outputs the three Sphere objects.

The following input ports are available:

**Maximum Iterations**

Suppose you want the node to output ten objects only, even though there are more than ten objects that can be found by the Iteration Path. Here you can set the maximum number of iterations.

**Object**

You can connect a reference object directly to this port. This saves you entering the path manually in the Reference Path box.

**On**

The On input port takes a Bool value that enables (True) or disables (False) the node. The node is enabled automatically if you do not add this port.

The following output ports are available:

**Count**

Outputs the total number of objects in the hierarchy being searched. In other words, this is the maximum number of objects that the node would be able to find via the Iteration Path.

**Object**

Outputs the objects found.
Material

This node searches for materials in the Material manager. The search begins with the top left material. Choose which type of material the node should search for in the Attribute manager (Material Type setting).

You can use the Material node to, for example, access a particular material in the Material manager. Here, the first material, then the second material, and finally the third material are passed on to an object. The Compare node controls the progress in the Material manager according to the current animation time. The materials are changed automatically over time.

In addition, you can exclude materials from the search by dragging them into the Exclude box in the Attribute manager.

The following input ports are available:

First Material
This port takes an integer value which defines where the search should begin. A value of 0 means the search will begin with the first material in the Material manager onwards. A value of 1 means the search will skip the first material and begin with the second material instead, and so on.

Maximum Materials
This port defines the maximum number of materials that the node will output. A value of 0 means the node will output all the materials that match the search criteria.

On
The On input port takes a Bool value that enables (True) or disables (False) the node. The node is enabled automatically if you do not add this port.
The following output ports are available:

**Count**
Outputs the total number of tags found. This value depends not only on the search criteria but also on the value of the Maximum Materials port. (The Maximum Materials value can limit the number of materials the node outputs, even if there are further materials of the desired type).

**Material**
Outputs the materials found.

**ObjectList**

When you want to introduce a known number of objects into the expression, this node is often the quickest way to do it (quicker than the Hierarchy node). In the Object manager, drag the names of all the objects that you want the node to output and drop them into the Iteration List box in the Attribute manager. Each time the node is called, the node will output these objects one after another, from the top of the list to the bottom.

The On input port takes a Bool value that enables (True) or disables (False) the node. The node is enabled automatically if you do not add this port. The Count port outputs the total number of objects in the Iteration List.

**Selection**

This node enables you to access Selection tags. The node outputs the index numbers of the points or polygons in the Selection tag one after another.
The following input ports are available:

**On**

The On input port takes a Bool value that enables (True) or disables (False) the node. The node is enabled automatically if you do not add this port.

**Tag**

Connect the desired Selection tag (point, polygon or edge) to this port.

The following output ports are available:

**Count**

Outputs the total number of elements in the Selection tag. In the case of edges, this value will actually be double the number of edges in the tag. This is because the node will output the number of points that make up the edges, not the number of edges.

**Iteration**

Outputs the index numbers of the points, polygons or edges in the Selection tag one after another. Edges are again a special case — for each edge, the node outputs the index numbers of its points.

**Tag**

This node outputs an object’s tags. Connect the object whose tags you want to access to the Object input port. In the Attribute manager, set Tag Type to the type of tag you want to access. You can exclude individual tags from the search by dragging them into the Exclude box in the Attribute manager.

Tag nodes enable you to access the various tags that belong to an object. Choose which type of tag you want to search for in the Attribute manager. Here, the Tag node searches for a Point Selection tag. Once found, the tag is passed to a Selection node.

The following input ports are available:

**First Tag**

Usually all tags that belong to the object are checked, starting with the first tag. However, this value enables tags to be skipped. For example, if First Tag is set to 0, the search begins with the first tag and all tags are checked. However, if First Tag is set to 1, the first tag will be skipped and the search will begin with the second tag.
**Maximum Tags**

Here you can specify the maximum number of tags that the node will output. For example, a value of 3 means that the node will output the first three tags only that match the search criteria. If the port receives the value 0, the node will output all tags that match the search criteria.

**Object**

Here, connect the object whose tags you want the node to output.

**On**

The On input port takes a Bool value that enables (True) or disables (False) the node. The node is enabled automatically if you do not add this port.

The following output ports are available:

**Count**

Outputs the total number of tags found. This value depends not only on the search criteria but also on the value of the Maximum Tags port. (The Maximum Tags value can limit the number of tags the node outputs, even if the object has further tags of the desired type).

**Tag**

Outputs the tags found.

---

**22 XPresso Editor**

**XGroups, p.916**

The following options and commands have been added to the Attribute manager.

**Active**

The nodes inside an XGroup will only be evaluated if the Active option is enabled. Alternatively, pass a Bool value of TRUE or 1 to the XGroup’s On port.

**Protect**

This button enables you to protect XGroups by means of a password. The contents will then be hidden. Only by entering the correct password can the user gain access to the contents of the XGroup.

To protect an XGroup, first lock its contents (from the context menu, enable View > Locked). In the Attribute manager, click the Protect button. In the dialog that opens, enter a password for the XGroup. The XGroup is now protected.
The procedure for unprotecting the XGroup is almost the same in reverse. First click the Unprotect button then enter the password into the dialog that opens. To make the contents of the XGroup visible once more, unlock the XGroup (disable View > Locked from the context menu) and choose the desired display mode (context menu: View). For more information on locking and unlocking XGroups, see Chapter 22, ‘XPresso Editor’, in the CINEMA 4D reference manual.

XPresso Context Menus

View, p.929

The description of the Standard display size is incorrect in the manual. The correct description is:

If you have changed the View setting of a node or XGroup to Minimized or Fullscreen, you can restore the node or XGroup to its normal size by choosing Standard. (The normal size of the node or XGroup is the size it had when last resized manually).

The following applies to protected XGroups:

If an XGroup has been protected with a password, the XGroup’s contents will remain hidden even after choosing the Locked command. To make the XGroup’s contents visible and editable once more, in the Attribute manager, click the Unprotect button and enter the correct password. Proceed with caution when protecting an XGroup. Should you forget the password, you won’t be able to access the XGroup’s contents.
1 Radiosity

Radiosity parameters in the Render Settings

Stochastic Mode, p.5

Accuracy

The Stochastic Mode can now be calculated adaptively as well, the advantage being that you can get the same (or better) result in less time. This algorithm achieves this by spending less time on unimportant parts of the rendering and more time on the critical parts of the scene (such as corners and touching objects). This is controlled dynamically by the Accuracy and Stochastic Samples settings.

With this new adaptive nature, it is now possible for critical parts of the scene to receive more rays than defined by the Stochastic Samples setting (which now acts as the base value).

➔ For a non-adaptive Stochastic Mode — i.e. the same mode as in previous versions — set Accuracy to 0.
Example

With the new adaptive Stochastic mode, this scene is rendered twice as quickly as before. Instead of using the old Stochastic Samples value of 64, the scene now arrives at the same result with a Stochastic Samples setting of just 32 (with Accuracy set to 80%).

Identical Noise Distribution

This option can also be used with the normal radiosity mode. However, you’ll barely see its effects with normal radiosity.

When using Stochastic mode, sometimes the rendered image can be very grainy (see picture below), especially when using a low value for Accuracy or Stochastic Samples. This type of grain is often called ‘noise’.

If Identical Noise Distribution is enabled, the noise will be identical (or at least very similar) for each frame of the animation. If on the other hand the option is disabled, the noise will be different for each frame. This can lead to better results, especially when using scene motion blur. Or without motion blur, it can generate an attractive effect, similar to looking through night vision binoculars.
HDRI

What exactly are HDRIs (High-Dynamic Range Images)?

HDRIs are ideal for use in combination with radiosity and to recreate very natural and realistic lighting situations. HDRIs also have a big impact on the realism of reflections because they can produce reflections on objects that are much brighter than reflections caused by normal textures.

In short, HDRIs are images with a very high brightness range, much higher than normal RGB images, which are limited to a brightness range of 8 bits (256 variations) per color. Standard RGB images have the limitation of the brightest possible color being a 255/255/255 white. If you use such an image in combination with Radiosity to illuminate a scene, even the brightest white won’t be bright enough to simulate a real lighting situation. The resulting renderings often look flat and lack contrast. That’s because in reality the difference between a dark light (e.g. candle flame) and a bright light (e.g. sun) is too big to be represented within an 8-bit RGB image.

However, using a HDRI, the sun can have a brightness value of 6000 which is taken into account by the Radiosity algorithm and when rendering reflections. This results in beautifully and realistically illuminated scenes. The pictures below illustrate the difference:

✔ Try re-rendering your old scenes using HDRI. You will be amazed by how much better they look.
HDRI Types

There are three main types of HDRI: HDRI Latitude/Longitude, HDRI Light Probes and HDRI Horizontal/Vertical Cross.

**HDRI Latitude/Longitude**

These images are distorted in such a way that they can be mapped onto a sphere (or a Sky object) using spherical mapping which produces perfect results. This type of HDRI works best in CINEMA 4D. Other types of HDRI can be converted to this projection type as described below.

**HDRI Light Probes**

These HDRIs are generated by taking pictures of a reflective sphere.

**HDRI Horizontal/Vertical Cross**

These HDRIs are laid out like an unfolded cube and are primarily meant to be mapped onto the inside of a cube. For CINEMA 4D however, we recommend using the Latitude/Longitude type.

**Usage**

At www.debevec.org/Probes/ you will find a number of HDRIs which you can use for testing.

HDRIs can be used as textures within the Color channel or the Luminance channel of a material. When using a Sky object, you can use it in either one of these. Only if you want to use an actual sphere as an environment you will need to use the luminance channel.

If you want the HDRI to be the only source of light in your scene, remember to switch off the Auto Light option in the Render Settings (on the Options page). Depending on the intensity of the HDRI texture, your scene may be too dark or too bright. There are two ways to adjust the HDRI brightness:

- The Strength parameter on the Radiosity page in the Render Settings and the Global Illumination Generate/Receive options on the Illumination page of the material with the HDRI texture.
or by directly changing the brightness of the HDRI texture:

- The Brightness slider on the Color or Luminance page (whichever one the HDRI is loaded into) of the HDRI material. The brightness can also be set higher than 100% by setting the Mix parameter to Multiply and by manually typing in a Brightness value. You should use this method if you are using the HDRI as a visible background image because the first method only changes the amount of light cast by the image, not the brightness of the image itself. The first method also doesn’t change the brightness for reflections.

The texture interpolation type also affects the brightness of the HDRI. MIP mapping for example softens the texture which enlarges the bright areas of a HDRI and results in brighter renderings. The interpolation types None and Circle result in darker renderings. However, they can cause the problems described in the following note.

➔ The extreme differences in brightness of HDRIs can result in “patchiness” when using Radiosity which is especially visible on flat surfaces such as a floor. In this case, using MIP mapping and increasing the O (Offset) value for the HDRI texture helps to reduce this effect. The Offset is used to soften textures and in this case it can be used to smooth out those areas of extreme brightness in a HDRI. The objects are much more evenly lit as a result.

However, when using this method, you will probably still want your reflections and background to be as crisp and sharp as possible. So it is a good idea to use two Sky objects — one for generating Radiosity and the other just for the reflections (and as a background). Both Sky objects need to have a Compositing Tag assigned. For the one which is used for Radiosity, disable the options Seen By Camera and Seen By Rays. For the one which is used for the reflections and as a background picture, disable the Seen By GI option.

**Convert HDR Cross, Convert HDR Probe**

Some HDRIs come as Probe or Cross projection types. To get the best result in CINEMA 4D, these should be converted to Latitude/Longitude (spherical) maps and projected onto a Sky object. For this purpose there are two plugins included — Convert HDR Cross and Convert HDR Probe — which are available from the Plugins > Advanced Render menu. Choosing one of these commands opens a file selector where you can select a HDR Probe (or Cross) image. The image is converted to a Latitude/Longitude map and is saved in the same folder with the extension ".con.HDR" added to the end of the file name. The result is displayed in the Picture Viewer.
3 Depth of Field Filter

When using the Depth of Field filter, artefacts may appear in the render, especially in the region of ‘Front Blur’. These artefacts are not a bug but a limitation of post effect blur. However, the filter has been enhanced to ensure better results.

**Lens Details**

**Lens Intensity, p.31**

Contrary to the description in the reference manual, this parameter does not define the brightness of the artefacts. Rather, it defines the brightness range within which the artefacts should appear.

5 Glow Filter

If you render part of the picture instead of the whole picture, you may get a different glow effect. The glow effect may also change if you adjust the antialiasing settings. This is because antialiasing affects the brightness of pixels, which in turn affects the glow. Please note that these are not bugs but limitations of post effect glow.

**Glow**

**Back Luminosity, p.46**

The caption is incorrect. It should read: Back luminosity set to a high value (bottom left) and a low value (top right).

**Glow Edges, p.47**

For glowing edges to function correctly, you must enable an Object ID and assign a Compositing tag to the glowing object. In addition, enable a buffer for the tag that is set to the same ID number.
Dynamics has been adjusted to the new CINEMA 4D R8 interface and several of its dialogs and tag parameters have moved to the Attribute manager. This makes animating these parameters easy and also allows them to be used in combination with XPresso.

The parameters of the following dialogs can now be found in the Attribute manager:
- Solver Object
- Gravity
- Drag
- Wind
- Rigid Body Dynamic Tag

The parameter names are the same:

Old and new dialog

The Solver Object, for example, still has the same page names and the parameters are still arranged in a similar way. The only exception is the Energy Loss parameter which has been moved from Details to Main.

The Basic and Coord. pages in the Attribute manager are of minor importance for dynamics. You can use the Coord. page to change the position, rotation and scale of modifiers like wind and gravity (as for any other object in CINEMA 4D), but using the Move, Scale and Rotate tools in the top toolbar will probably still be quicker in most cases.
Solver Object

Integration Method, p.187

Two new integration methods have been added:

Euler

This method is the least accurate but also the fastest. It is about half as accurate as Midpoint but roughly twice as fast and should be used whenever simple collisions between many objects need to be calculated and as long as soft bodies are not involved.

An example to illustrate this:

In this experiment, a sphere is shot into a field of many other spheres, resulting in thousands of collisions. The time needed for the calculation (with a similar result and using an oversampling value of 2) is as follows:

Euler: 77%
Midpoint: 100%
Runge-Kutta: 186%

These values may differ depending on the scene, but they illustrate how much of a difference the integration method can make. Note that the values include the dynamics calculation as well as the actual rendering.
Softbody

Soft bodies often took a long time to calculate in previous versions but now a new integration method has been added specifically for soft bodies. When using this method, only small Oversampling values are needed — in most cases a value of 2 or 4 is enough. Soft body collisions especially can now be calculated much faster. Depending on the scene, the calculation can be up to 100 times faster than using a different integration method (which would also require higher Oversampling).

✔ Try experimenting with the Dynamics Tutorial scenes to see how the new Softbody integration method compares with the other methods.

Rigid Bodies

Initialization

Initialize Scene, p.223

This command has been renamed to Initialize All Objects.

Keyframe Animation

Baking a Dynamics Animation

Bake Scene, p.257

This command has been renamed to Bake All Solvers.

Clear Scene, p.257

This command has been renamed to Clear All Solvers.
Nodes

**TP Initiator Group**

**PPass, p.15**

A new output port has been added:

**Particle Count**

Outputs the total number of particles in the particle group.

**PPass AB, p.16**

Two new output ports have been added:

**Particle Count A, Particle Count B**

These ports output the total number of particles in the two particle groups.

**New Nodes**

**PBlurp (TP Generator)**

This new node breaks up an object into fragments, moves the fragments along a path, then reassembles (morphs) them to form a completely new object (the target object). For example, you can morph the 3D text ‘Thinking’ into the 3D text ‘Particles’.

The following input ports are available:

**Animation Phase**

This input port controls the progress of the morphing effect. A value of 0% means the original object is fully intact, a value of 100% means the morphing is complete and the target object has been fully assembled from the fragments.
Animation Time

Because the node’s parameters can be keyframe animated, by default the CINEMA 4D time is used internally to ensure that the values are interpolated correctly. However, you can pass your own time value to this port. This should be of the data type Time, which is a Real number in the simplest case. If no value is passed to this port, CINEMA 4D’s time is used.

On

The On input port takes a Bool value that enables (True) or disables (False) the node. The node is enabled automatically if you do not add this port.

The following output ports are available:

Fragment Count
Outputs the number of fragments that have been produced by the node.

Fragment Number
Outputs the internal number of the fragment particle that is currently being generated.

Fragment Particle
Outputs the fragment particles that are currently moving between the objects. Using this output port they can be assigned to individual particle groups or shapes.

Remaining Count
The object is fragmented gradually. This port outputs the number of fragments yet to be created.

Remaining Number
Outputs the internal numbers of all the remaining fragment particles.

Remaining Particle
Outputs the fragment particles which have not yet been created and which are not yet moving towards the next object.
The main controls of the PBlurp node are found in the Attribute manager:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>× Objects</td>
<td>Cube, Sphere</td>
</tr>
<tr>
<td>Animation Phase</td>
<td>0 %</td>
</tr>
<tr>
<td>Stay</td>
<td>0 %</td>
</tr>
<tr>
<td>Frag</td>
<td>100 %</td>
</tr>
<tr>
<td>Next</td>
<td>100 %</td>
</tr>
<tr>
<td>In Tangent</td>
<td>100</td>
</tr>
<tr>
<td>Out Tangent</td>
<td>100</td>
</tr>
<tr>
<td>From To Type</td>
<td>2 to 42</td>
</tr>
<tr>
<td>Angle</td>
<td>45 °</td>
</tr>
<tr>
<td>Radius</td>
<td>50 %</td>
</tr>
<tr>
<td>Count</td>
<td>5</td>
</tr>
<tr>
<td>Thickness</td>
<td>1 %</td>
</tr>
</tbody>
</table>

**Objects**

In the Object manager, drag the objects that should decay into fragments or be formed from the fragments and drop them into this box in the Attribute manager. These objects must be polygon objects. Primitive objects must be converted to polygons before being used with this node.

You can use two or more objects. These can be completely different shapes if you wish and they can have a different number of polygons. The top most object in the Objects list will be used as the starting object for the effect. This object will then fragment and morph into the second object in the list, which in turn will fragment and morph into the third object (if present), and so on.

A spline curve is used to control the path between the objects. Each object has virtual tangents which you can adjust to control this curve. The Left Tangent and Right Tangent parameters define the length of each tangent arm. There is a further way to control the spline curve: you can add Null objects to the Objects list. Using these nulls, you can adjust the course of the spline between two polygon objects (the positions of the nulls will adjust the shape of curve).

**About the other parameters...**

You can set the following parameters separately for each object in the Objects list. Select the object whose parameters you want to edit, then adjust the parameters as desired.

**Stay**

By default, the time it takes an object to break up into fragments is the same as the time it takes for the fragments to form the next object. However, if you set Stay to a value higher than 0%, the decay into fragments will be delayed.
A value of 100% corresponds to the length of time an object needs to decay into another object. Therefore if you increase the Stay value, the object will have less time to break up into fragments. In other words, if you increase this value, the object will break up into particles more rapidly (after an initial delay).

**Frag**

As mentioned above (see ‘Stay’), by default the time it takes for an object to decay into fragments is the same as the time it takes for the next object to be formed out of the fragments. However, the Frag value enables you to control how quickly the object decays into fragments. A value of 0% means the object will decay into fragments immediately, while a value of 100% means the decay will take the full time available.

**Next**

This value controls how long it will take for the fragments to move between the two objects. A value of 0% means the particles will move immediately to the position of the target object, ignoring the spline curve that is between the objects. In contrast, a value of 100% means the fragments will take the full time available to move to the position of the target object.

**In Tangent, Out Tangent**

These values control the lengths of the tangents for the object currently selected in the Objects list. A value of 0 leads to a linear spline path between the objects. Greater values lead to paths that are more curved.

**From To**

This setting controls the direction in which the object decays into fragments. For example, a setting of -Y To +Y means the object will decay from the negative Y-axis in the direction of the positive Y-axis. The object’s axis system is used.

**Type**

Here you can control the shape and appearance of the fragments. You can use your own objects as fragments — connect the Fragment Particle port to a PShape node.

Choose one of the following modes:

*Single Faces*

Each face of the object breaks up into one fragment.

*Smooth And Distance*

This mode activates the Angle and Radius settings. The number of fragments created depends on the direction of the surface normals. Neighboring surfaces that point in a similar direction (as defined by the Angle value) are grouped together to form a single fragment. The Radius value controls the size of these fragments. The Radius value is a percentage where 100% represents the total size of the object. A value of 50% therefore leads to fragments that are half the size of the object that is decaying.
Count
This mode activates the Count parameter, which enables you to specify the exact number of fragments that the object will break up into.

Angle
This setting is available only if Type is set to Smooth And Distance. It defines the maximum difference in angle between normals of neighboring surfaces that will give rise to a single fragment.

Radius
This setting is available only if Type is set to Smooth And Distance. The value defines the size of the fragments relative to the size of the object.

Count
If Type is set to Count, here you can enter the total number of fragments that the object is broken up into.

Thickness
This parameter gives the fragments thickness by extruding them. The strength of the extrusion is defined as a percentage, where 100% corresponds to the object’s greatest dimension. Keep in mind that adding thickness to the fragments means that the effect will require more surfaces to be created, and hence more RAM will be needed.

No Fragments
This option hides the fragments (for some compositing tasks, it can be useful for the fragments to be invisible but for the holes to still appear in the object).

Remaining Type
Here you can define the appearance of the object once fragments have been broken away from it.

None
The original object disappears completely as the fragments break off.

Hollow
Holes are carved out of the object as the fragments break off. This gives the effect that the object is breaking up into its component parts.

Solid
Each fragment that breaks off takes a chunk out of the object. This is comparable to the peeling of an orange. In contrast to the Hollow mode, the object remains solid — there are no unfilled holes, just chunks missing.
**Thickness**

In contrast to the Thickness value previously described, here you define the thickness of the object that remains behind. If Remaining Type is set to Hollow, you can define a thickness for the object so that it still appears to be solid after fragments have broken off. Keep in mind that adding thickness to the object requires more surfaces and hence more memory. For best results, set both Thickness parameters to the same value.

**Restricting materials to invisible selections**

The PBlurb node automatically creates invisible selections for the fragments and the parts left behind. You can restrict textures to these selections to apply different materials to the corresponding parts. To do this, select a Texture tag and in the Attribute manager, type the name of the desired invisible selection into the Selection box. Note that the names of these invisible selections (listed below) are case-sensitive. For example, if the texture should be applied to the edges of fragments, type FEDGE into the Texture tag’s Selection box.

- **FEDGE**
  The edges of fragments.

- **FBACK**
  The backfaces of fragments.

- **FREDGE**
  The edges of the remaining parts of the object.

- **FRBACK**
  The surfaces that are under the fragments. If Remaining Type is set to Solid, these surfaces will become visible. If Remaining Type is set to Hollow, these surfaces form the insides of the object that is left behind.

**Using the PBlurb node**

To morph one object into another using the PBlurb node:

- Drag-and-drop two objects into the PBlurb node’s Objects list in the Attribute manager. For both these objects, set Remaining Type to Hollow. In most cases, you should hide these objects.

- Create a Particle Geometry object (Plugins > ThinkingParticles > Particle Geometry).

- Animate the PBlurb node’s Animation Phase parameter. You can animate it directly in the Attribute manager if you wish using the context menu. For example, at frame 0, record an Animation Phase value of 0%, then at frame 60, record an Animation Phase value of 100%.
PDraw (TP Generation)

With the help of a virtual brush, the PDraw node enables you to freely draw particles in a 3D space. Once you have drawn the particles, the node will act as a container for them.

The following input ports are available:

**Animation Time**

Because the node’s parameters can be keyframe animated, by default the CINEMA 4D time is used internally to ensure that the values are interpolated correctly. However, you can pass your own time value to this port. This should be of the data type Time, which is a Real number in the simplest case. If no value is passed to this port, CINEMA 4D’s time is used.

**On**

The On input port takes a Bool value that enables (True) or disables (False) the node. The node is enabled automatically if you do not add this port.

The following output ports are available:

**Birth Count**

Outputs all the particles created during the current frame. If you want to access all particles and not just the newly born ones, use a PPass node instead.

**Birth Num**

Outputs the number of particles born during the current frame.

**Particle Birth**

Outputs the number of the last particle born during the current frame.

The following parameters are available in the Attribute manager:

**Type**

Use this menu to define where the particles are drawn relative to the position of the mouse pointer.

**Point**

The particles are drawn is the same position as the mouse pointer. The values for Radius and Count will be ignored in this mode. The number of particles created will depend on how quickly and for how long you move the mouse pointer.
Spherical

The particles are drawn inside an imaginary sphere which surrounds the mouse pointer. The Radius value defines the size of this sphere and the Count value defines the number of particles drawn per mouse movement.

Radius

When Type is set to Spherical, the particles are drawn inside an imaginary sphere which surrounds the mouse pointer. Here you can define the size of this imaginary sphere. You can also adjust the size of the sphere interactively in the viewport — see ‘Drawing the particles’, which follows later in this chapter.

Count

Defines the number of particles drawn per mouse movement. This parameter has no effect when Type is set to Point (in which case, one particle will be drawn per mouse movement).

Draw Position

This parameter currently has no effect.

Life Time, Life Variation

The Life Time parameter defines how long particles live for before they die. You can vary the lifetime of the particles using the Life Variation value.

Random Seed

The node calculates variations in the lifetime of the particles based on this starting value. PDraw nodes that have identical settings will generate exactly the same variation unless you give them different Random Seed values.

Remove

To delete all the particles in the PDraw node, click this button. All of the node’s particles will be deleted, not just those that are currently visible.
Drawing the particles

To draw particles in a view:

- Select the PDraw node.
- Choose Plugins > ThinkingParticles > TP ParticleDraw.
- Move the time slider to the frame at which the particles should be born.
- Ctrl-drag to draw the particles.
- If Type is set to Spherical, you can Shift-drag to change the size of the virtual sphere inside which the particles are drawn.
- Once you have finished drawing the particles for this frame, move to the next frame at which you want particles to be born. Draw the particles for the new frame.
P MatterWaves (TP Generator)

With this node, you can use lights or textures or both to control the emission of particles from object surfaces. Note that the node will only influence the particles at the time of their birth.

The PMatterWaves node is controlled primarily in the Attribute manager. The following parameters and options are available there:

**Object**

Into this box, drag the object that should emit the particles. This must be a polygon object. If you want to use a primitive or NURBS object instead, you must first convert it to polygons.

**Selection**

Here you can restrict the emission of particles to specific polygons. Create a Polygon Selection tag and in it store a selection for the polygons that you want to emit particles. Drag the Selection tag into this box.

**UVW**

If you want to control the particle emission using a texture, the object must have a UVW tag. If the object has only one UVW tag, you can leave this box empty (the sole UVW tag will be used automatically). If on the other hand the object has multiple UVW tags, drag the UVW tag that you want to use into this box.

**UVW Type**

- **Clamp**

  In this mode, particles will only be emitted from parts of the surface that are covered by the UVW map.

- **Unclamped**

  In this mode, particles can be generated from anywhere on the object, even if the picture covers only part of the surface.

**Lights**

If you want lights to control the particle emission, drag the lights from the Object manager and drop them into this box.
Creating particles

Birth Type, Count, Rate, Shot

Count

In this mode, the Count value defines the maximum number of particles that may exist at any one time. The Count value works closely together with the Life value. For example, if Life is set to 5 and Count is set to 100 particles, the number of particles will increase steadily to 100. As particles continue to die, new particles are emitted to keep the count at 100.

Rate

In Rate mode, the Rate value defines the number of particles that are born per second of the animation — the total number of particles will therefore be affected by the frame rate.

Shot

In Shot mode, Shot defines the number of particles created per frame of animation.

Birth Type

Constant

The particles will be emitted at a constant rate from anywhere on the object.

Texture

This mode enables you to control particle emission using a picture or 2D shader. To load a picture, click the Image button and use the dialog that opens to locate and load the picture. To load a 2D shader, click the arrow button that is next to the Birth Texture box and choose the desired shader from the menu that opens. Note that it is the brightness of the picture or shader that will control the emission rate. Color plays no part, therefore you can use grayscale pictures if you wish.
The picture or shader will be fitted to the object’s surface using the chosen UVW coordinates. Parts of the surface that receive a white color from the picture or shader will emit particles at the maximum rate, while parts of the surface that receive a black color will emit no particles at all.

The Birth Gradient enables you to control how the birth of particles is affected by brightness values between the two extremes of black and white. The left edge of the gradient represents the parts of the picture or shader that are black, while the right edge represents the parts of the picture or shader that are white. With the default gradient, a gray value will cause particles to be emitted at a rate that is relative to its brightness. For example, a gray color with 70% brightness will cause particles to be emitted at 70% of the maximum rate.

You can adjust the gradient by adding new knots or by moving the existing ones or both. For example, if you drag the black knot to the right until all of the gradient is black except for a small, white band on the right, the particles will only be emitted in those areas where the color value is very close to the maximum white tone.

A gradient from black to white does not represent a smooth increase in the particle emission rate. Rather, together with the Birth Gradient, you are merely defining threshold values for the full emission rate.

**Light**

Parts of the object that are fully lit by this light will emit particles at the maximum rate, while parts of the object that receive little or no illumination from this light will emit no particles. As with the Texture mode, you can use a gradient to adjust how intermediate brightness values affect the emission rate.

**Texture & Light**

In this mode, you can use lights and a picture or shader to control the emission rate. A surface will then only emit particles at the maximum rate if it is fully illuminated by the lights and it receives a white color from the texture or shader.

**Lifetime of the particles**

**Life**

Here enter how long the particles should live for, in frames.
**Life Variation**

This value enables you to vary the lifetime of the particles that are emitted (all of the node's variation values are based on the Random Seed value).

**Life Type**

*Constant*

The lifetime of the particles is independent of lights and textures.

*Texture*

This mode enables you to control the lifetime of the particles using a picture or 2D shader. To load a picture, click the Image button and use the dialog that opens to locate and load the picture. To load a 2D shader, click the arrow button that is next to the Life Texture box and choose the desired shader from the menu that opens. Note that it is the brightness of the picture or shader that will control the lifetime. Color plays no part, therefore you can use grayscale pictures if you wish.

The picture or shader will be fitted to the object's surface using the chosen UVW coordinates. Parts of the surface that receive a white color from the picture or shader will emit particles that live for the maximum lifetime, while parts of the surface that receive a black color will emit particles that die immediately.

The Life Gradient enables you to control how the lifetime of the particles is affected by brightness values between the two extremes of black and white. The left edge of the gradient represents the parts of the picture or shader that are black, while the right edge represents the parts of the picture or shader that are white. With the default gradient, a gray value will give particles a lifetime that is relative to its brightness.

You can adjust the gradient by adding new knots or moving the existing ones or both. For example, if you drag the black knot to the right until all of the gradient is black except for a small, white band on the right, the particles will die almost immediately except in those areas where the color value is very close to the maximum white tone.

*Light*

You can control the lifetime of particles using one or more light sources. Parts of the object that are fully lit by this light will emit particles that have the maximum lifetime, while parts of the object that receive little or no illumination from this light will emit particles that die immediately. As with the Texture mode, you can use a gradient to adjust how intermediate brightness values affect the lifetime.

*Texture & Light*

In this mode, you can use lights and a picture or shader to control the lifetime. A surface will then only emit particles that live for the maximum lifetime if it is fully illuminated by the lights and if it receives a white color from the texture or shader.
Speed of the particles

Speed
Here you can enter the speed of the particles.

Speed Variation
This value enables you to vary the speed of the particles that are emitted (all of the node’s variation values are based on the Random Seed value).

Speed Type

Constant
The speed of the particles is independent of lights and textures.

Texture
This mode enables you to control the speed of the particles using a picture or 2D shader. To load a picture, click the Image button and use the dialog that opens to locate and load the picture. To load a 2D shader, click the arrow button that is next to the Speed Texture box and choose the desired shader from the menu that opens. Note that it is the brightness of the picture or shader that will control the speed. Color plays no part, therefore you can use grayscale pictures if you wish.

The picture or shader will be fitted to the object’s surface using the chosen UVW coordinates. Parts of the surface that receive a white color from the picture or shader will emit particles that travel with the maximum speed, while parts of the surface that receive a black color will emit particles with zero speed.

The Speed Gradient enables you to control how the speed of the particles is affected by brightness values between the two extremes of black and white. The left edge of the gradient represents the parts of the picture or shader that are black, while the right edge represents the parts of the picture or shader that are white. With the default gradient, a gray value will give particles a speed that is relative to its brightness.
You can adjust the gradient by adding new knots or moving the existing ones or both. For example, if you drag the black knot to the right until all of the gradient is black except for a small, white band on the right, the particles will travel with zero speed unless emitted from areas where the color value is very close to the maximum white tone.

**Light**

You can control the speed of particles using one or more light sources. Parts of the object that are fully lit by this light will emit particles that have the maximum speed, while parts of the object that receive little or no illumination from this light will emit particles that have zero speed. As with the Texture mode, you can use a gradient to adjust how intermediate brightness values affect the speed.

**Texture & Light**

In this mode, you can use lights and a picture or shader to control the speed. A surface will then only emit particles that are travelling at full speed if it is fully illuminated by the lights and if it receives a white color from the texture or shader.

**Size of the particles**

**Size**

Defines the size of the particles.

**Size Variation**

This value enables you to vary the size of the particles emitted (all of the node’s variation values are based on the Random Seed value).

**Size Type**

- **Constant**

  The size of the particles is independent of lights and textures.
Texture

This mode enables you to control the size of the particles using a picture or 2D shader. To load a picture, click the Image button and use the dialog that opens to locate and load the picture. To load a 2D shader, click the arrow button that is next to the Size Texture box and choose the desired shader from the menu that opens. Note that it is the brightness of the picture or shader that will control the size. Color plays no part, therefore you can use grayscale pictures if you wish.

The picture or shader will be fitted to the object’s surface using the chosen UVW coordinates. Parts of the surface that receive a white color from the picture or shader will emit particles of the maximum size, while parts of the surface that receive a black color will emit particles of zero size.

The Size Gradient enables you to control how the size of the particles is affected by brightness values between the two extremes of black and white. The left edge of the gradient represents that parts of the picture or shader that are black, while the right edge represents the parts of the picture or shader that are white. With the default gradient, a gray value will give particles a size that is relative to its brightness.

You can adjust the gradient by adding new knots or moving the existing ones or both. For example, if you drag the black knot to the right until all of the gradient is black except for a small, white band on the right, the particles will be of zero size unless emitted from areas where the color value is very close to the maximum white tone.

Light

You can control the size of particles using one or more light sources. Parts of the object that are fully lit by this light will emit particles that are the maximum size, while parts of the object that receive little or no illumination from this light will emit particles that have zero size. As with the Texture mode, you can use a gradient to adjust how intermediate brightness values affect the size.

Texture & Light

In this mode, you can use lights and a picture or shader to control the size. A surface will then only emit particles that are of the maximum size if it is fully illuminated by the lights and if it receives a white color from the texture or shader.
Distance of particles from the object

Distance
Here enter how far away particles should be born from the object’s surface. This distance is measured in the direction of the surface normals.

Distance Variation
This value enables you to vary the distance from the object’s surface at which the particles are born (all of the node’s variation values are based on the Random Seed value).

Distance Type
Constant
The distance from the surface at which the particles are emitted is independent of lights and textures.

Texture
This mode enables you to control the distance of the particles from the object using a picture or 2D shader. To load a picture, click the Image button and use the dialog that opens to locate and load the picture. To load a 2D shader, click the arrow button that is next to the Distance Texture box and choose the desired shader from the menu that opens. Note that it is the brightness of the picture or shader that will control the distance. Color plays no part, therefore you can use grayscale pictures if you wish.

The picture or shader will be fitted to the object’s surface using the chosen UVW coordinates. Parts of the surface that receive a white color from the picture or shader will emit particles at the maximum distance from the surface, while parts of the surface that receive a black color will emit particles directly from the surface.

The Distance Gradient enables you to control how the distance of the particles from the object is affected by brightness values between the two extremes of black and white. The left edge of the gradient represents the parts of the picture or shader that are black, while the right edge represents the parts of the picture or shader that are white. With the default gradient, the gray tones will emit particles whose distance from the surface is relative to the tone’s position between the two color extremes.
You can adjust the gradient by adding new knots or moving the existing ones or both. For example, if you drag the black knot to the right until all of the gradient is black except for a small, white band on the right, the particles will only be of a reasonable distance from the surface if emitted from areas where the color value is very close to the maximum white tone. Particles emitted from other areas will be emitted very close to or directly from the surface.

**Light**

You can control the distance using one or more light sources. Parts of the object that are fully lit by this light will emit particles that are the maximum distance from the surface, while parts of the object that receive little or no illumination from this light will emit particles directly from their surface. As with the Texture mode, you can use a gradient to adjust how intermediate brightness values affect the distance.

**Texture & Light**

In this mode, you can use lights and a picture or shader to control the distance. A surface will then only emit particles from the maximum distance away if it is fully illuminated by the lights and if it receives a white color from the texture or shader.

**Direction of the particles**

![Direction of the particles](image)

**Direction**

This value will only be used if you are using a texture or lights to define the direction of the particles as they leave the surface of the object. The Direction value will then be used as a type of multiplier in order to determine the influence of the texture or light.

**Direction Variation**

This value enables you to vary the direction of the particles (all of the node’s variation values are based on the Random Seed value).

**Direction Type**

*Normal*

The direction of the particles is independent of lights and textures. The particles will travel in the direction of the surface normals.
**Texture**

This mode enables you to control the particles using a picture or 2D shader. To load a picture, click the Image button and use the dialog that opens to locate and load the picture. To load a 2D shader, click the arrow button that is next to the Direction Texture box and choose the desired shader from the menu that opens. Note that it is the brightness of the picture or shader that will control the particles. Color plays no part, therefore you can use grayscale pictures if you wish.

The picture or shader will be fitted to the object’s surface using the chosen UVW coordinates. For the calculation of the direction vector, in each case the neighboring brightness values of the point considered will be analyzed. This corresponds to the process that is used to calculate bump effects. Therefore the direction of a particle will only differ from the direction of the normal if different brightness values are next to each other.

The Direction Gradient enables you to control how the particles are affected by brightness values between the two extremes of black and white. The left edge of the gradient represents that parts of the picture or shader that are black, while the right edge represents the parts of the picture or shader that are white.

**Light Direction**

You can use the direction of the light to control the particles. Drag the lights that you want to use into the Lights box.

**Light Reflection**

In this mode, the direction of the light reflected by the object is calculated. As with the Light Direction mode, drag the lights that you want to use into the Lights box.

In addition, the following input ports are available:

**On**

The On input port takes a Bool value that enables (True) or disables (False) the node. The node is enabled automatically if you do not add this port.

**Animation Time**

Because the node’s parameters can be keyframe animated, by default the CINEMA 4D time is used internally to ensure that the values are interpolated correctly. However, you can pass your own time value to this port. This should be of the data type Time, which is a Real number in the simplest case. If no value is passed to this port, CINEMA 4D’s time is used.

The following output ports are available:

**Birth Count**

Outputs all the particles born during the current frame. If you want to access all particles and not just the newly created ones, use a PPass node instead.

**Birth Num**

Outputs the number of particles born during the current frame.
**Birth UVW**
Outputs the UVW coordinates used for each particle born, as a vector.

**Particle Birth**
Outputs the number of the last particle born during the current frame.
NET RENDER

- Although NET Render can create RLA and RPF files, these file formats do not contain 3D data.
- NET Render can render up to a maximum of 9,999 frames per animation.