

---

## Rocket Launch

Maya's particle system is a powerful and extensive collection of tools and features for simulating a variety of effects such as dust, sand, fire, clouds, explosions, smoke, and liquids. Effects such as these are found in countless feature film and broadcast shots and have become part of every day effects work for many in the industry.

This tutorial explains how to set up and render a rocket launching. The focus is placed on setting up the smoke trail of the rocket using software particles.

The following topics are discussed:

- What are particles in Maya?
- Creating particles and emitters
- Parenting an emitter to animated geometry
- Adjusting particle attributes
- Adding fields to affect particle motion
- Keyframing emission rate in the Graph Editor
- Changing particle radius with respect to particle age
- Assigning shaders to particles with Hypershade
- Rendering considerations.



---

**Note:** **Before you start:** This tutorial requires Maya scene files. If you haven't installed the tutorial files go to the "Try Maya" section of the "Taste of Maya" CDROM to install.

**Starting Maya:** You can start Maya by double-clicking the **Maya 2.5 Evaluation** icon on the desktop or from **Start → Programs → Maya 2.5 Evaluation → Maya**.

**In Maya:** Once the tutorial files have been installed you will need to set the current project in order to access the Maya scene files. To set the current project from within Maya select **File → Project → Set...** and Navigate to the directory where you installed the Maya scene files. Select the directory: *Maya\_tutorial\_data* and press **OK**.

---

## Rocket Trail

To create the smoke trail in the above image three different particle objects that each have slightly different motion and shading characteristics. All three particle objects will be emitted from one emitter that is parented to the rocket.

**"thick" particles:** provides base layer of particles for chunky appearance

**"softer" particles:** provides a layer of wispieness

**"trail" particles:** used to simulate the streaking effect in the smoke trail.

Once you tune the motion and emission of the particles using fields and animation curves, lighting and shading will be added then you will render.

## What is a Particle Object in Maya?

Particles in Maya differ from geometry in the following ways:

- Particles are points in space. They require special handling at render time because they do not contain surface information.
- Particles can be rendered using hardware or software rendering methods. The particle Render Type attribute controls which of these two methods is used. The rocket example uses software render types only. Software render types render slower but allow for details such as shadows, detailed texturing, and reflections.
- In Maya, Individual particles belong to a common collection called the *particleShape* object. Just as CVs of a geometric object belong to their shape node, individual particles can be thought of as components to the **particleShape** node.

## Creating the Thrust Emitter

A particle emitter is like a cannon that fires particles into space.

### 1 Open scene file and playback

- **File → Open;** select *rocket\_start.ma*

This file contains a rocket on a launch pad. The rocket launch has already been keyframed. Playback to see the rocket launching.

### 2 Add a directional emitter

- Press **F4** to switch to the Dynamics menus
- Use the Outliner to select *thrust* (within the rocket group)

- Select **Particles** → **Add Emitter** → , and set the following:

**Emitter Name** to **thrustEmit**;

**Emitter Type** to **Directional**;

**Max Distance** to **5**

**Min Distance** to **1**

**Rate** to **45**;

**Spread** to **0.15**;

**Direction X, Y, and Z** to **-1, 0, 0** respectively;

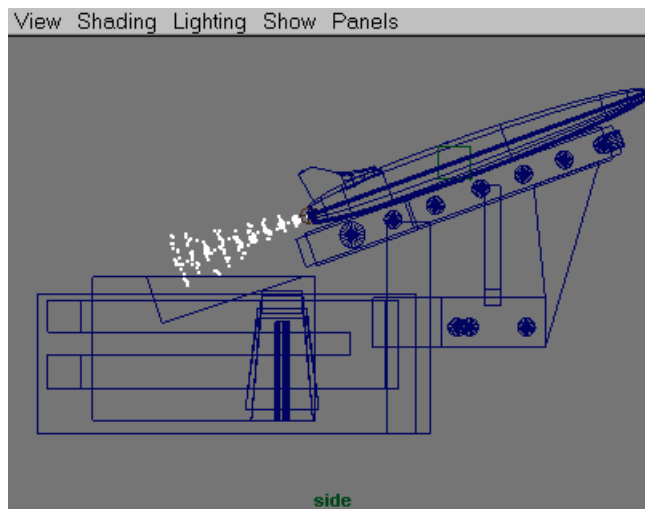
**Speed** to **150**;

- Press **Add** then **Close**

Min and Max distance determine a distance from the emitter in which emission can occur. The values are used here keep the particles from overlapping with the tail of the rocket. Rate is the number of particles per second emitted, Speed controls how quickly the particles leave the emitter.

### 3 View and rename what was created

- Press play on the timeline to watch the particles emit.



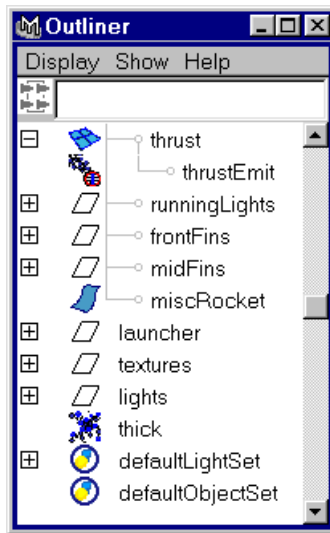
*Particles trailing behind the rocket*

---

**Note:** Make sure **Playback Speed** is set to **Free** in **Options** → **General Preferences** → **Animation**. Otherwise, the emitter may appear to “sputter” particles instead of providing a constant emission.

---

- Open **Window** → **Outliner** to see the newly created particle object called *particle1* and *thrustEmit* emitter.
- Double click on *particle1* in the Outliner and rename it *thick*



*Outliner showing emitter and associated particle object*

- Select the particle object and set **Level of Detail** to **0.25** in the Channel Box.

Level of detail controls what percentage of the particles will be displayed on the screen and in the final render. Lowering this value lets you quickly evaluate motion while testing. Remember to set this back to 1 when rendering. Using level of detail is very important to working with particles efficiently.

---

**Tip:** To create a “stand-alone” emitter that does not get placed within an object hierarchy you can use **Particles** → **Create Emitter**. This emitter can still be parented to an object later if desired.

---

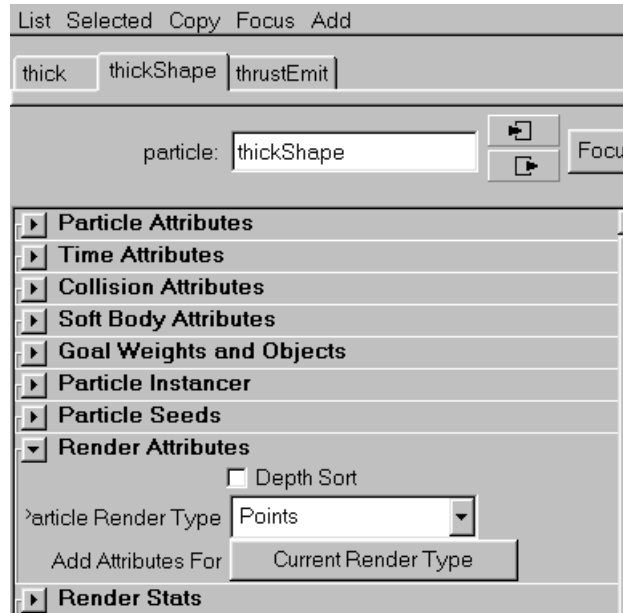
## Adjusting particle attributes

Now that the emitter is in place, you can begin to make some preliminary changes to the particles so they are easier to see and so they behave more like a smoke trail would.

### 1 Open the Attribute Editor for thick particles

The Attribute Editor window for the *thickShape* object is where most particle attributes are adjusted. Changes made here affect every particle component that belongs to the currently selected particle object.

- Continue on from the last step or open the file *rocket\_thick.ma*
- Select *thick* in the Outliner
- **Ctrl - a** to open the Attribute Editor



*Attribute Editor window for thick particles*

## 2 Set the Particle Render Type and associated attributes

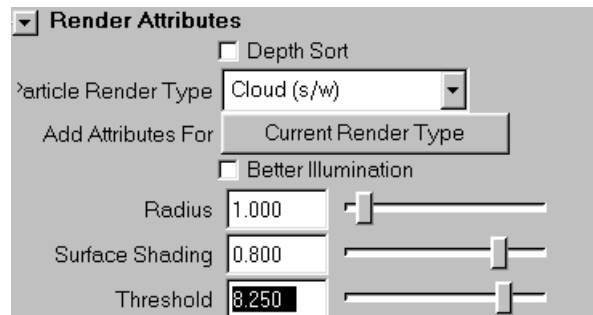
- Set **Particle Render Type** to **Cloud** (under **Render Attributes**)
- Click the **Current Render Type** button

A list of attributes associated with the cloud render type is displayed. Change the following settings:

**Surface Shading** to **0.8**

**Threshold** to **8.250**

These settings control some rendering properties of these clouds that will be more important later on. You will not see any change in the perspective window.

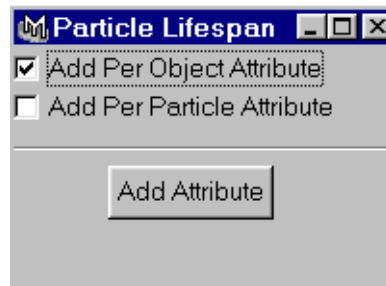


*Setting thick particles to cloud and adjusting some attributes*

## 3 Add per object lifespan

Adding a lifespan attribute allows you to control how many seconds each particle will be displayed in the scene.

- Press the **Lifespan** button under **Add Dynamic Attributes**
- Choose **Add Per Object Lifespan Attribute**



- Set **lifespan** to **5** in the **Render Attributes** section of the Attribute Editor

## Adding Fields

Fields act like forces to move the particles around in different ways. Here you will add gravity, turbulence, and radial fields and connect them using three different methods.

### 1 Create gravity and connect it using Connect/Add

Here you will create a “stand-alone” field and connect the field to the particles so they fall more realistically.

- Make sure nothing is selected
- Choose **Fields** → **Create Gravity**
- In the Outliner select the *thick* particles then **Ctrl-select** the *gravityField1*
- Choose **Connect/Add** → **Connect to Field**

This connects the gravity field to the particles. If you switch to wireframe mode (press 4) and select only *gravityField1*, the particles highlight in pink to indicate a connection is present.

### 2 Playback and adjust particle conserve attribute

As you playback, the particles quickly fall instead of just floating. To obtain a look of ejecting and easing to a stop, you can adjust the amount of momentum the particles conserve each frame.

- Open the Attribute Editor for *thickShape*
- Adjust the **Conserve** attribute near the top of the Attribute Editor.

A value of **0.1** works well.

This means that each frame each particle keeps only 10 percent of the velocity it had in the previous frame. Try experimenting with other conserve values.

### 3 Create turbulence and connect it using dynamic relationships

Now you will apply some turbulence so the particles scatter a little bit and will use the Dynamic Relationships Editor to connect the field. This editor can be used for managing many types of dynamics connections including fields, collisions, and emission.

- Rewind to **Frame 1** and make sure nothing is selected
- Choose **Fields** → **Create Turbulence** - □

Set the following:

**Magnitude** to **100**

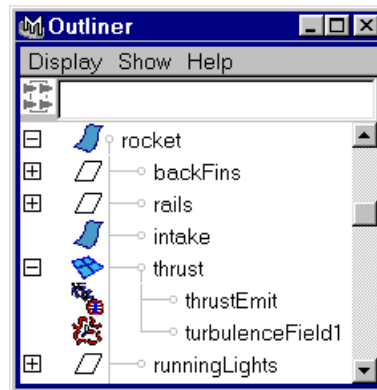
**Attenuation** to **1**

Press **Create** then **Close**

An Attenuation of 1 causes the field influence to diminish over distance causing particles further away from the field to be less turbulent.

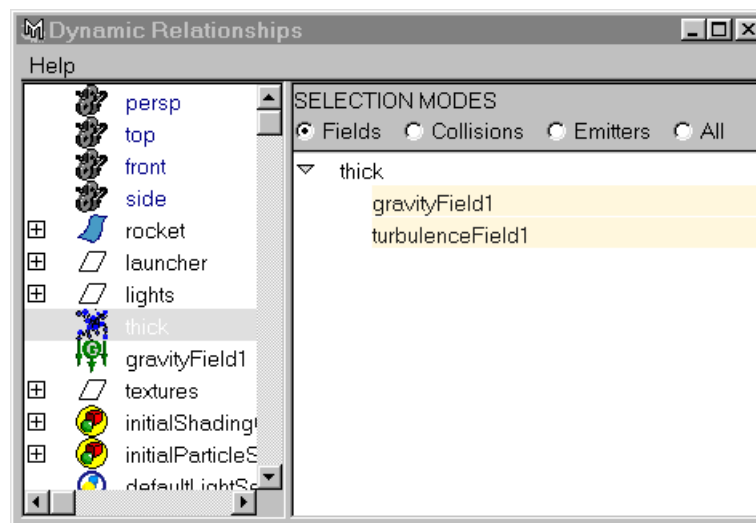
- Press **W** to invoke the **Move** tool then translate the field's icon to roughly the same location as the *thrustEmit* object (near the back end of the rocket).
- With the field still selected, **Ctrl-select** the *thrust* geometry object in the Outliner (not the emitter), then **Edit** → **Parent**

This causes the field to follow along with the rocket.



*Outliner showing parenting of turbulence field*

- Open **Windows**→**Relationship Editors** → **Dynamic Relationships...**
- Make sure the **Selection Mode** in this Editor is set to **Fields**
- Highlight *thick* on the left side and *turbulenceField1* on the right
- Notice that *gravityField1* is also highlighted confirming it is connected as expected.



*Dynamic Relationships Editor showing gravity and turbulence connected*

#### 4 Add a radial field to push the particles outward.

The next step is to make the particles expand slightly after they have left the rocket. This can be achieved by adding a radial field.

- Select *thick* then choose **Fields** → **Add Radial** - □

Set the following options:

**Magnitude to 0.8**

**Attenuation to 0.5**

**Apply Per Vertex to On**

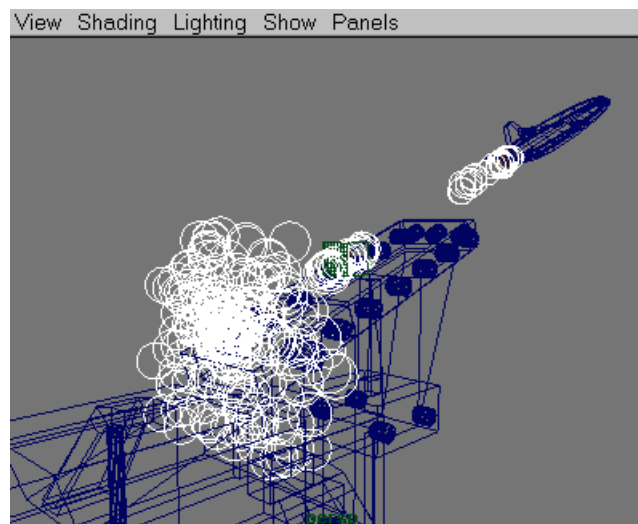
**Use Max Distance to On**

**Max Distance to 8**

Press **Add** then **Close**

Add Radial is used here so the field is automatically parented to the particles. Turning apply per vertex on causes each individual particle to behave as though there is an independent radial field attached to it. This keeps all the particles pushing away from each other.

- Use Window **Relationship Editors** → **Dynamic Relationships** to connect *thick* particles to *radialField1*.
- Playback to test the motion.
- The file *rocket\_with\_fields.ma* is a set up version of this scene up to this point.



*Particles pushed away from the center by a radial field*

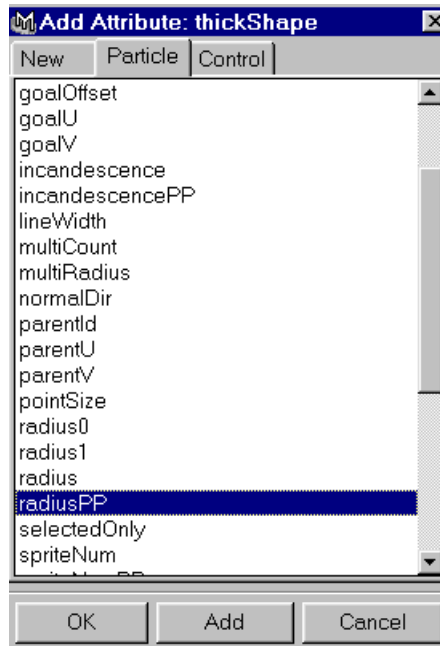
## Changing Each Particle's Radius Over Time

Currently every particle in *thick* has the same radius (1.0) You can give each particle its own radius value and change that radius value over the course of the particle's lifespan by adding a per particle radius attribute (*radiusPP*) that is controlled by a ramp. This way you can make the particles exit small, then grow, then get smaller again by the time they die, similar to how puffs of smoke behave.

### 1 Add *radiusPP* to *thick* in the Attribute Editor

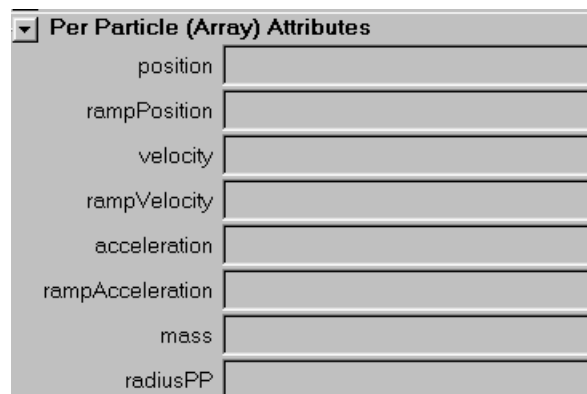
- Select *thick* and open the Attribute Editor
- Click the *thickShape* tab at the top of the Attribute Editor
- Click the **General** button under **Add Dynamic Attributes**
- Click the **Particle** tab on the resulting window and select ***radiusPP*** then press **OK**





*Adding a radiusPP attribute*

**Note:** The particles disappear because radiusPP is currently 0. In cases such as this where a radius and radiusPP attribute are present, the radiusPP values override the radius attribute.

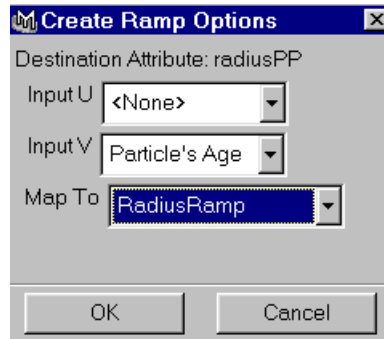


*radiusPP shown in thick's Attribute Editor*

## 2 Assign a pre made ramp to thick's radiusPP

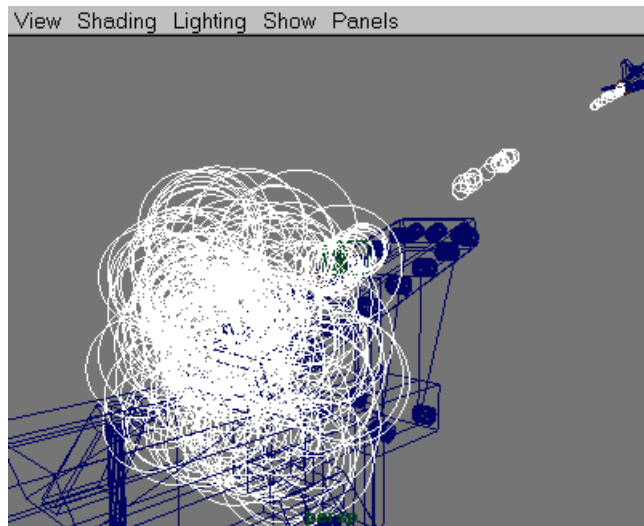
Since radiusPP is currently 0, you need to assign a value to it so the particles grow and shrink. Assigning a pre-made ramp is a quick way of doing this. A ramp is just a gradation between values.

- In the Attribute Editor for *thick*, **RMB** in the field next to **radiusPP**, and select **Create Ramp** - ☐



*Selecting a premade ramp from the Create Ramp Options Box*

- Set **Map To** to **RadiusRamp**. This is a gradation of values that has been set up for you so the radius grows and shrinks at the right times. Leave **InputU** and **InputV** as they are. Then press **OK**
- Rewind and playback. The file *rocket\_radiusPP.ma* contains a setup version of the scene up to this point.



*Particles with changing radius values*

## The Takeoff Burst

A rocket exudes more smoke at initial takeoff. Therefore, you will need to animate the emission rate on the *thrustEmit* emitter to simulate this.

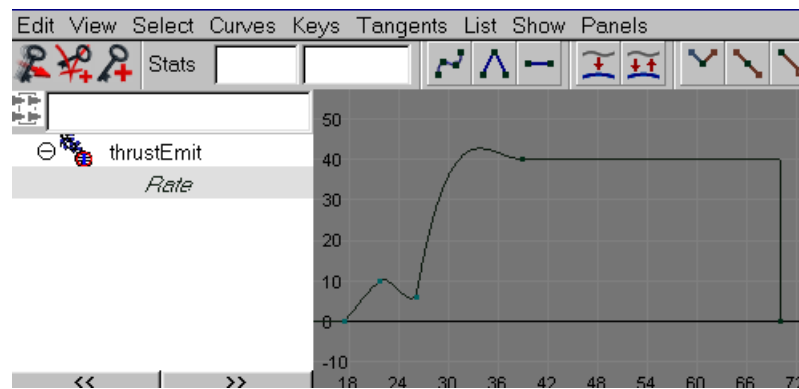
### 1 Open file or continue on from above steps

- *rocket\_radiusPP.ma* can be used as a starting point

### 2 Keyframe *thrustEmit* rate

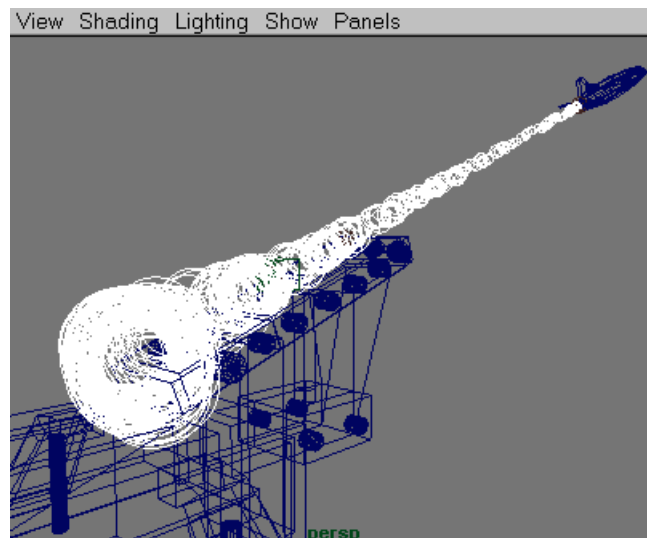
- Select **Windows** → **Saved Layouts** → **Persp/Graph/Outliner**
- Select *thrustEmit* and open the Channel Box
- Use the timeline to go to frame **18**
- Set rate to **0** in the Channel Box
- **RMB** click on **rate** in the Channel Box, choose **Key Selected**

- This keyframes the rate to 0 at frame 18
- Set 4 more keyframes using the values below.
    - frame 22: rate = 10
    - frame 26: rate = 6
    - frame 39: rate = 40
    - frame 70: rate = 0
  - The goal is to have an animation curve in the Graph Editor that looks similar to the picture below. Note that the last two keyframes in the picture below use stepped tangents. For more information about adjusting keyframe tangents and weights in the Graph Editor refer to the on-line documentation.



*Animation curve in the Graph Editor controlling emission rate*

**Tip:** If there are too many particles for you to see what is happening or if the playback is too slow you can temporarily lower your emission rate in the Graph Editor or lower the **Level Of Detail** attribute on the particle object. Temporarily setting the Render Type to Points will also help.



*The result of the keyframed emission rate*

## Rendering

Now the focus will shift to getting these particles to render like smoke. You will use a scene file that has some additional particle objects added to it and which has lighting set up for you. Your task is to assign the appropriate shaders to the particles.

### 1 Open file and analyze contents

- Open *rocket\_render.ma*

This file contains 2 additional particle objects that are emitting from the same emitter as the thick particles.

**softer:** These particles are slightly larger than the *thick* particles

**trail:** Tube render type particles whose length is determined by how fast they are moving. These particles will create trailing streaks behind the rocket.

- There are display layers for each particle object. Use these display layers to toggle the visibility of the particles on and off. This helps you to manage the number of particles displayed on the screen and aids in speeding test renders. The launcher has been templated to speed test renders.

### Render speed considerations

- You will be doing some test renders in with this scene. Keep in mind that the speed of the render will be greatly affected by the number of particles and the percentage of screen space those particles occupy. It is best to choose framings for test rendering that keep this in mind. Also, don't forget to use the level of detail attribute and the display layers in your scene to help manage the speed of your renders.

### Assigning Shaders for the Particles

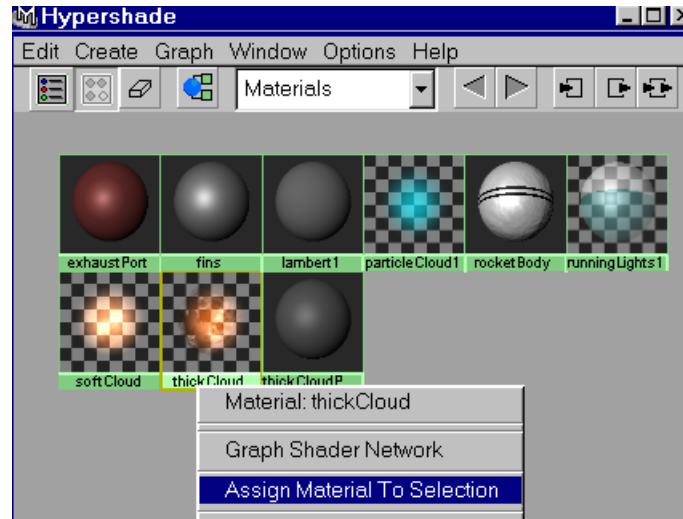
You will use two volumetric particle cloud shaders that are provided within the scene file.

- **thickCloud:** Provides a chunky edge appearance to the thick particles. This will be applied to *thickParticles*
- **softCloud:** More transparent than thickCloud. Provides a softer boundary overlaying the thickCloud and also tones down the brighter highlights found in the *thickCloud*. This will be assigned to *softer* and *trail* particles.

### 2 Using Hypershade

The Hypershade window is where you build and assign shaders in Maya.

- Open **Windows** → **Outliner** and select *thick*
- **Shift-select** the *thickClouds* shader in the Hypershade window
- **RMB** on the text label of *thickCloud* shader and select **Assign Material to Selection**
- Repeat the process for *softer* and *trail* except assign the *softCloud* shader to both of them instead.

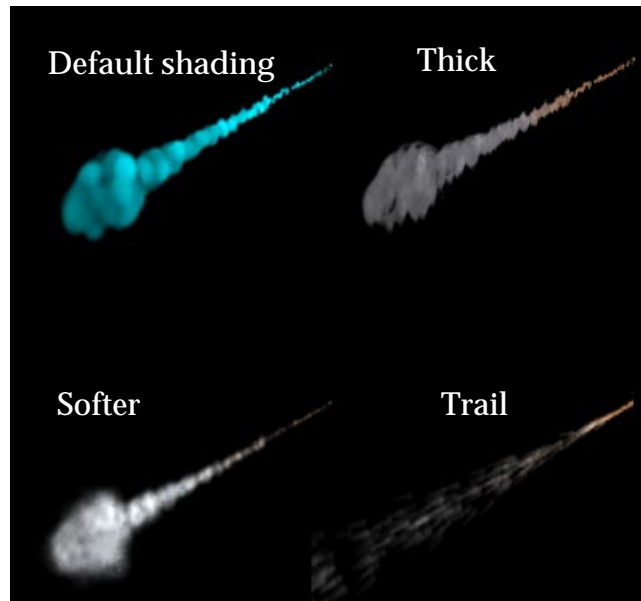


*The RMB menu for assigning shaders in Hypershade*

### 3 Test Rendering individual layers

To get an idea of what each shader and particle object is contributing to the image, use the display layers to display only one particle object at a time then render a frame. Generally when adjusting shaders, it is best to focus on one particle object at a time. This keeps rendering times lower and helps you to see what is going on.

- Make sure the layer bar is displayed (**Options** → **Layer Bar**)
- Play up to frame **40**,
- Hide *softer* and *trail* using the layer bar  
**RMB** in the Layer Bar on *softerLayer*, *trailLayer* and toggle **visible** off.
- In the Render View Window choose **Render** → **Render** → **Current (persp)**
- Repeat the process (two particle layers hidden, one visible) so you have rendered one still image of each particle object.
- Finally, test render a frame with all three particle objects visible so you can see how they appear when combined.



*A small render of each particle object and the default particle shader*



*The three particle layers combined.*

---

**Tip:** Press the 1:1 button in the Render View window to view the Render at the accurate aspect ratio.

---

## Final Rendering Considerations

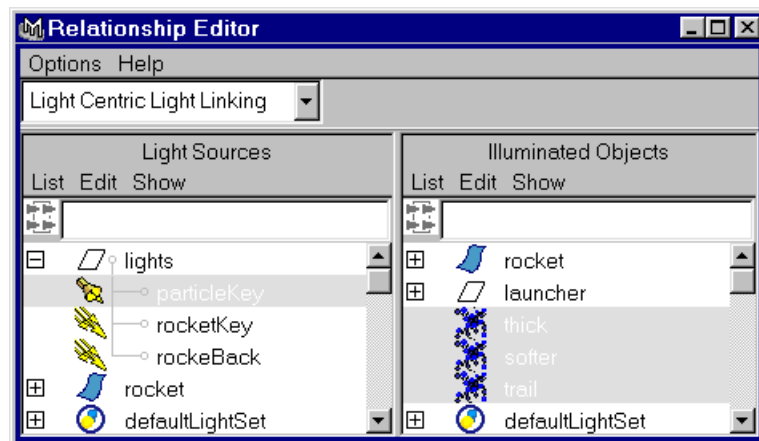
You have now completed the steps in this tutorial. This final section discusses some other options that are important to consider when rendering this type of shot in Maya. Rendering in Maya is done using **Windows** → **Render View**.



## Lighting

The lighting in this scene is set up so the particles are lit independently of other objects. Two directional lights illuminate the rocket and one spotlight illuminates the particles. There is also a colored point light that is parented to the back of the rocket.

- Lights can be created using the **Lights** menu under the Rendering menu set. Press **t** with the light selected to invoke special manipulators.
- A spotlight with linear decay, a slight color tint, wide cone, and high intensity work well for this shot.
- Setting **Illuminates by Default** to **OFF** on the lights allows you to set up light linking relationship so certain lights only shine on certain objects. Light linking is done using **Window** → **Relationship Editors** → **Light Linking** → **Light Centric**. Pick the desired light on the left and the objects you want it to illuminate on the right.



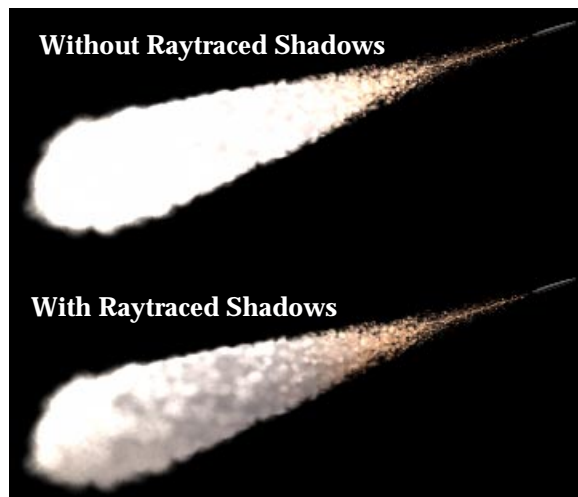
*Setting up light linking*

- Experiment with the placement of lights in the scene or try creating some of your own to see how they affect the rendered image.
- **Raytraced Shadows:** In order for the particles to have self shadowing the scene must be raytraced with raytraced shadows on the lights. To enable raytraced shadows select the light and enable raytraced shadows in the shadow section of the Attribute Editor. This is another benefit to linking lights. You can use depth map shadowing (which render faster) on lights linked to geometry in the scene.

### Render Globals Settings

The Render Globals window (**Windows** → **Render Globals**) is where many important rendering controls are set. For the rocket scene, the following settings are especially important to note. You should open the Render Globals window so you can see where these settings are found and experiment with them to see the results they produce.

- **Raytracing:** Enabled under Raytracing Options, required to obtain particle self-shadowing as mentioned above. Raytracing a scene slows things down considerably since much more computation is being done. Therefore, it is best to leave raytracing off until you are tuning the particle shadows for the final look.



- In the above image, The upper smoke image is lit from above and rendered without raytracing. The lower smoke trail was raytraced with raytraced shadows, it contains shadows within the smoke volume.
- **Motion Blur:** 2D motion blur adds realism for the rocket motion. Note that particles do not get motion blurred. Keep this off when test rendering.
- **Anti-Aliasing Quality:** Set to 3D Motion Blur Production when rendering the final shot, use Preview quality when doing test renderings.
- **Shadows obey light linking:** Located in the Render Options section, this option must be enabled so shadows from particles do not cast onto the rocket (the rocket is lit by different lights).
- The scene file *rocket\_final.ma* contains all of the above render global settings and is set up for a final quality render pass.

### Better Illumination

Better Illumination is a checkbox option found on the particle object just below the render type setting in the Attribute Editor. Enable this option when doing final renders only. This option is for cloud particles



only and increases the quality of the lighting on the particles and helps to prevent artifacts and shadow popping.

## **CONCLUSION**

You have now seen how a typical software particle workflow session is done in Maya. To recap, this tutorial covered the following areas:

- Creating particles and emitters
- Parenting an emitter to animated geometry
- Adjusting particle attributes
- Adding fields to affect particle motion
- Keyframing emission rate in the Graph Editor
- Changing particle radius with respect to particle age
- Assigning shaders to particles
- Rendering considerations.