

Chapter 21

Particle Effects

by Brandon Davis

A *particle system* is a procedural animation tool in 3ds max 4 that enables users to animate large groups of objects by a series of rules and procedures. Unlike the



traditional keyframe method of animating objects by interactively adjusting them from pose to pose, procedural animation is automated by the system and defined by a user set of parameters, such as velocity, mass, orientation, and so on. These parameters can be keyframed and are constantly evaluated by the system and applied to the objects or particles in this case. Particle systems are effective at replicating the motion of natural phenomena, such as rain, fire, and smoke. Confused? Think of it this way: Animating a stream of raindrops one by one would be very tedious. However, with a particle system, you can quickly animate hundreds of raindrops by just determining a few parameters and letting the system create the motion.

In this chapter, you explore the available particle tools in 3ds max 4 and how they work. Then you employ particle systems to create two effects: a turbulent, trailing pixie dust effect and a confetti effect. Specifically, this chapter discusses:

- The basic components of particle systems
- Particle classes
- Common particle parameters
- Simple Particles: Spray and Snow
- Super Particles: SuperSpray, Blizzard, PCloud, and PArray
- Special particle maps
- Space warps for particles

**Note**

One of the first, if not *the* first, uses of particle systems was for the Project Genesis formation shot in the 1982 film *Star Trek II: The Wrath of Khan*. This minute-long shot took the fledgling Industrial Light & Magic computer graphics group nearly six months to complete and was the first fully computer-generated sequence for a feature film. Bill Reeves contributed his particle system fire to the highly procedural project, which was led by Alvy Ray Smith and included other legendary CG pioneers, such as Loren Carpenter, Rob Cook, and Tom Duff. Incidentally, this is the group that would eventually split off in 1986 and become Pixar.

Basic Components of Particle Systems

At the base level, particle systems consist of three basic components: an emitter, particles, and fields. An *emitter* is often a nonrendering object that determines the space where particles are created or born. Some emitters are single points, and others are surfaces or volumes that contain particles. One of the main differences among the particle systems available in 3ds max 4 is the way they handle emitters.

Particles in their simplest form are actually just infinitely small points in space that contain only position and rotation information. Many particle systems offer particle primitives that are geometry types that can be automatically applied to particles. These often are simple shapes, such as spheres, cubes, tetrahedrons, and single polygons. At this level they are often employed for simple effects, such as sparks and rain. Some particle systems can reference complex geometry that is even hierarchically animated. This technique is often used for creating procedural animation for crowds or flocks of objects, such as birds.

Fields are an important component of particle systems that are used to influence a particle from the point of birth to death—essentially the entire lifetime of a particle. Examples of fields are Gravity, Wind, and Deflectors. Some fields' effects are localized to an area of influence, and others are global in scope. By using fields, particle systems can be made to act naturally (smoke and fire) or unnaturally (lasers and tractor beams).

All particles go through the following cycle: birth, life, and death. *Birth* is used to describe the formation of a particle in space. Birth is normally a function of time and quantity. In 3ds max, particles are born at either a variable rate per frame or from a set total. In the case of a total, a particle pool is created based on a value, and the particle count in the scene never exceeds that amount. All particles have a *life* value that determines how long they will be present in the simulation. When this value has been reached, *death* occurs and they are no longer a participant.

Although particles are alive, they can be influenced by fields. For example, Drag may be applied to particles to realistically dampen their motion over time. This is not applied at birth or death, but instead over the course of a particle's lifetime. All three stages can be controlled directly within the particle system. This is where many of the similarities end, however.

Particle Classes

In 3ds max 4, there are three classes of particle systems: Simple, Super, and Third-Party.

Simple Particles systems were the first ones included in Max 1 and these include Spray and Snow. They are considered simple because they contain only a few parameters and are designed for the most basic effects. Spray and Snow are still integrated and interactive, making them a big step up from their ancestors, the AXPs of 3d Studio DOS.

Super Particles systems were added in Max 2 by Yost Group members Eric and Audrey Petersen. This extended particles a bit further and in the process completely decentralized them. SuperSpray, Blizzard, and PCloud are nearly identical, except for the way they emit particles. SuperSpray emits particles from a point in space directionally in a stream that can be spread out conically or thin and wide like a fan. Blizzard emits from an invisible rectangular surface, and PCloud emits from within a geometric volume.

PArray is the only Super Particles system that differs more than any other. PArray emits particles from the geometry of objects and has the most options for determining what those particles are composed of. Unique to PArray is instanced geometry and fragments. Instanced geometry references an object in the scene to use for particles. Examples of such would be complex geometry such as detailed rocks or animated hierarchies such as a bird made up of several objects. Object fragments are created by placing particles on the surface of an emitter object, and then using that object's geometry to create fragments. A simple example would be to generate fragments from a cube. Each triangle polygon forming the cube is detached and becomes particle geometry. Where the particles move, the fragments move. This is a simple method that retains a few parameters to control how many fragments are created and how thick they are. Unfortunately, because they're working exclusively with triangle polygons, their results tend to be synthetic looking.

Third-Party particle systems are easily integrated into 3ds max 4 via its extensible architecture. Most Third-Party particle systems are specialized and operate uniquely compared to one another.

One common production add-on to the existing particle systems in 3ds max is Particle Studio from Digimation, an event-driven particle system that enables a non-linear and flexible particle workflow. In this method of working, particle systems are organized into blocks of events, the timing of which can be determined relatively, absolutely, or by particle age. Oleg Bayborodin (www.orbaztech.com), the developer of Particle Studio, has effectively balanced speed, ease of use, flexibility, and capability into a particle system for 3ds max 4.

Cebas Computer's (www.cebass.com) Matterwaves particle system is unique because it contains bitmap-driven parameters. Users can, for example, use a bitmap applied to emitter geometry to determine the rate of emission. This can be slow to work with, but has its merits. Cebas also recently released its flagship event-driven particle system, Thinking Particles. This is a highly procedural approach, building on the

workflow method of Particle Studio—or Side Effects’ Houdini, for that matter—but also greatly expanding upon it. My first impressions are that this is an overly complex particle solution, though, it’s probably the most capable and flexible currently available to 3ds max users.

Next Limit’s (www.realflow.com) RealFlow is a standalone computational fluid dynamics simulator that has hooks into 3ds max 4. Because it treats particles as fluids in a medium and allows for particle-to-particle interaction, RealFlow can simulate substances such as viscous fluids and gasses realistically.

**Note**

Prior to Max 1 in 1996, particle systems were rudimentary in 3d Studio DOS. Most (if not all) were third-party products, and only a few gave any sort of method of visualization or feedback. The AXP plug-ins tended to be in a small window with a few parameters to describe the particle system. Users had to render frames to see how the particles looked and moved.

Common Particle Parameters

Each class of particle system in 3ds max 4 has a common set of parameters that are used to control aspects of the particle system.

The following sections describe the parameters common to each class of particle system, both Simple and Super.

Simple Particles: Spray and Snow

Spray and Snow share most parameters; Snow particles have rotation parameters and Spray particles do not. The following are parameters for the Simple Particles:

- **Viewport and Render Count.** These parameters control the total number of particles that are emitted over time with regard to viewport display and render display. The main purpose for these related parameters being uncoupled is that the user can work faster in the viewports by displaying only a portion of the total particle count, while still rendering with a full quantity.
- **Drop Size.** This parameter controls the scale of the particle geometry.
- **Speed and Variation.** These are a bit misleading. Speed determines the velocity of the particles at birth, and Variation not only varies the speed value but also implements a sort of spreading pattern, with higher values increasing the spread.

- **Tumble and Tumble Rate.** These are rotational parameters added to Snow. Tumble controls the amount of rotation of the particles and Tumble Rate controls the speed at which the particles rotate.
- **Render Tetrahedron and Facings.** These parameters are the given particle primitives or predetermined geometry for the particles. Tetrahedrons are shaped like simple raindrops, and Facings are square polygons.
- **Start and Life.** These parameters control particle birth and death. Start determines the start frame of particle emission in absolute time, and Life is a relative value that determines how many frames a particle is “alive.” Life is relative, because particles are constantly born and emitted, starting on the Start frame and continuing forever, so a particle born at frame 100 with a life of 60 dies at frame 160.
- **Birth Rate and Constant.** These parameters are used to pulse or burst emit particles, a simple method that is expanded by the Super Particles. Birth Rate determines the amount of particles emitted in a given round. When activated, the Constant check box disables the Birth Rate parameter.

Super Particles: SuperSpray, Blizzard, PCloud, and PArray

The Super Particles class of particle systems share many common parameters. In fact, SuperSpray, Blizzard, PCloud, and PArray are primarily differentiated by how they handle emitters. The following are parameters for the Super Particles:

- **Icon Size and Emitter Hidden.** These parameters are cosmetic values that control the size of the emitter icon and determine whether it is hidden or displayed.
- **Dots, Mesh, Ticks, and Bbox.** These parameters are the viewport display methods for particles. Dots vary depending on the viewport display method used (*Heidi dots* are single-pixel and OpenGL dots are much larger). Mesh displays the actual geometry to scale in the viewports, a method that tends to be computationally intensive but still useful. Ticks, like Dots, are constant in size with the exception of being shaped like crosses. Bbox displays the bounding box of the particle geometry. This is often faster than using the Mesh method.
- **Percentage of Particles.** This parameter controls the percentage of actual particles displayed in the viewports. This does not affect the quantity of

particles at render time, but is used to speed up display by reducing the number of particles the display has to render.

Particle Generation

The Super Particles class was added in release 2 and builds on the previous particle systems, Spray and Snow. The main difference between the four Super Particle systems is the way in which they emit particles. Common particle generation parameters are as follows:

- **Use Rate and Use Total.** These parameters are the primary methods for controlling the amount of particles emitted. Use Rate determines a variable number of particles that are emitted at each frame. Use Total defines a total quantity of particles that the system can emit over time, the rate of which depends on the emission start and stop time.
- **Speed and Variation.** These parameters control the emission velocity of the particles. Variation controls the percentage at which Speed will be varied on a particle-by-particle basis. You use Variation as a common parameter with many particle systems because it's useful to break up the uniformity of values by adding some randomness to them.
- **Emit Start and Stop.** These parameters are the set frame numbers at which particles will start to be emitted and will stop.
- **Display Until and Life.** These parameters are used to determine when particles will die. Display Until is a rudimentary way of doing this absolutely so that on a given frame all particles will die. Life is like Spray, a relative value determining how many frames a particle will be present after it's been emitted.
- **Size, Grow For, and Fade For.** These parameters control particle geometry scale. Size is a multiplier of scale and has a Variation parameter tied to it to randomize the scale on a particle-by-particle basis. Grow For and Fade For are two parameters that cause particles to grow from a scale of zero to the full Size value and then fade to zero over a given set of frames.
- **Seed.** This parameter is an important random value that controls the uniqueness of particle systems. It's a random number that can be changed using the New button next to it. If multiple particle systems use the same Seed value, they emit particles identically. Use different Seed values to cause variation among particle systems.

Particle Types

The Super Particles class of particle systems shares three main types of geometric representations: Standard, MetaParticles, and instanced geometry. PArray contains a fourth type called Object Fragments.

Standard, MetaParticles, and instanced geometry determine the types of geometry the particles will be displayed as:

- **Standard.** This type offers several particle primitives, such as Triangle, Cube, Facing, and Sphere.
- **MetaParticles.** These are a special type of geometry primitive that is useful in creating amorphous liquid shapes. MetaParticles create spheres at each particle location, and then fuse the geometric surfaces together when their proximity to one another passes a user-defined threshold. When MetaParticles are sparsely spaced they appear as simple spheres, but when packed closely together they fuse into blob shapes. This is computationally expensive and should be used with extreme care.
- **Instanced Geometry.** This is a useful particle type that enables the user to reference separate objects in the scene as particle geometry. For example, a single teapot object can be referenced and will cause instances of the teapot to be emitted as particles. The instanced geometry can also be hierarchically animated, enabling effects such as swimming fish and flocking birds.

Rotation and Collision

All the Super Particles share the same set of Rotation and Collision controls, which specify the speed of rotation and starting angles. Collision is a bit misleading, because it has to do with particle-to-particle collisions only, not with objects in the scene. The following are Rotation and Collision controls:

- **Spin Time and Phase.** These control the amount of frames it takes for a particle to fully rotate, with low values causing faster spinning. Phase determines the starting angles of rotation.
- **Random, Direction of Travel, and User Defined.** These are three methods for controlling rotation. Random rotates the particles on varied axes. Direction of Travel forces the particles to orient themselves in the direction they are traveling, with an optional Stretch value to actually stretch the particle geometry based on velocity. User Defined sets an absolute rotational axis in X, Y, and Z.

- **Interparticle Collisions.** This is a set of parameters to test particle-to-particle collisions. This is computationally intensive and useful only in rare instances. *Calc Intervals Per Frame* is a parameter for controlling the rate at which the particle collisions are tested. *Bounce* is a percentage of force applied at collision.

Object Motion Inheritance

Object Motion Inheritance controls how much velocity is passed from the emitter to the particles. No motion inheritance causes the particle's speed to control velocity; 100% inheritance causes the full velocity of the emitter to be added to the particle emission speed. This is relevant only to a moving emitter. Object Motion Inheritance is a useful control for adding realism to particle motion. Unfortunately, the default value of 100% tends to confuse people. There are two controls:

- **Influence.** This determines what percentage of the particles is affected.
- **Multiplier.** This determines how much of the velocity is inherited from the emitter. A multiplier of 0 implies no inheritance, and 1.0 passes 100% of the emitter velocity to the particles.

Bubble Motion

Bubble Motion includes another set of parameters that are common to the Super Particles class of particle systems. These parameters are used to impart a wobbling motion to particles. Unfortunately, they aren't particularly intuitive and often require good viewport feedback to properly visualize. There are three main parameters to control Bubble Motion:

- **Amplitude.** This parameter controls the distance at which the particle will wobble away from its original direction.
- **Period.** This parameter controls the cycle time of particle oscillation, which is similar to waveform frequency.
- **Phase.** This parameter is used here similarly as it was with rotation, controlling the initial wobble offset.

Particle Spawn

All Super Particles have common controls for particle spawning. *Particle spawning* is the emission of particles from particles. There are several methods by which particles can be spawned over time:

- **None.** This method causes zero particle spawning.
- **Die after Collision.** This method causes particles to die when they collide with a Deflector space warp.
- **Spawn on Collision.** This method performs the opposite of the previous method, causing particles to be born on collision.
- **Spawn on Death.** This method causes a sort of fireworks effect by creating particle spawns when a particle dies.
- **Spawn Trails.** This method is pretty self-explanatory, causing particles to spawn trails of new particles behind them over their lifetime.

Each of the spawning controls has common parameters used to control the events and the spawned particles:

- **Spawns.** This parameter determines the number of particle spawns that happen over a particle's lifetime.
- **Affects.** This parameter determines the percentage of particles that will produce spawning.
- **Multiplier.** This parameter is a value that multiplies the number of particles spawned at a given event.
- **Directional Chaos.** This parameter is used to vary the direction at which particles are spawned.
- **Speed Chaos.** This parameter is similar to Directional Chaos, but instead varies the speed at which particles are spawned on a frame-by-frame basis.
- **Scale Chaos.** This parameter is similar to Directional Chaos and Speed Chaos, but varies the scale of spawned particles.
- **Lifespan Queue.** This parameter is used to determine several alternate lifetimes for spawned particles. Values are created and stored in the queue and then passed on to spawned particles on an event-by-event basis.
- **Object Mutation Queue.** This parameter works with instanced geometry and causes spawn particles to switch between instanced geometry types referenced in the queue on an event-by-event basis.

Parameters Unique Among Super Particles

As mentioned earlier, the main differences between the Super Particles are the methods used to emit particles. These unique features are described in the following sections.

Blizzard

Blizzard includes the following values:

- **Width and Length.** These are used to determine the size of the rectangular emitter object.

SuperSpray

SuperSpray includes the following values:

- **Off Axis and Spread.** These are two related values that control the angle of emission from a single point in space. Off Axis determines the angle off axis from Z (straight up through the emitter), similar to the elevation of a gun in a turret. Spread controls the angle at which particle emission is varied in a fan direction.
- **Off Plane and Spread.** These are two more related values to control the angle of emission from a point. Off Plane controls the angle of emission rotated around the plane of the emitter, similar to the azimuth of a gun in a turret. Spread controls the conical spread angle along the axis of emission. A value of 180 creates a complete cone shape.

PArray

PArray is unique because it uses geometry for emitters and can explode that geometry into fragments to use as particle geometry:

- **Object-Based Emitter.** This is a pickbutton used to determine which object will be used for emission.

There are several particle formation methods:

- **Over Entire Surface.** This method creates particles evenly over the surface of the emitter geometry. This is the most commonly useful method and thereby the default.
- **Along Visible Edges.** This method snaps particle emission to the visible edges of the emitter object.
- **At All Vertices.** This method restricts emission to the vertices of the emitter object.
- **At Distinct Points.** This method isolates particle emission to a given set of distinct points on the emitter's surface.

- **At Face Centers.** This method emits particles from the centers of all faces on the emitter object.
- **Use Selected Sub-Object.** This method is a check box that applies to all the previous methods. This enables the user to further restrict emission to only the selected sub-objects (faces, edges, verts) of the emitter geometry.

PArray has some unique parameters and methods for creating object fragments for particle geometry:

- **Thickness.** This parameter determines the size of the fragments' extrusion.

The following three methods determine how an object will fragment:

- **All Faces.** This method forces each face of the emitter geometry to be used as a polygonal fragment.
- **Number of Chunks.** This method sets a minimum number of fragments created.
- **Smoothing Angle.** This method controls the number of fragments created based on the angle of the surface normals of the emitter object. The higher the angle, the fewer the fragments created.

PCloud

PCloud is unique because it emits particles from within a volume instead of a surface. There are four methods to define the volume:

- **Box Emitter.** This method is a variable box shape controlled by length, width, and height.
- **Sphere Emitter.** This method is a variable spherical shape.
- **Cylinder Emitter.** This method is a variable cylindrical shape with parameters similar to Box Emitter.
- **Object-Based Emitter.** This method uses selected geometry from the scene to use as a volume emitter.

PCloud has unique particle motion methods:

- **Random Direction.** This method emits particles along random vectors.
- **Enter Vector.** This method uses X,Y, and Z vectors to determine the exact emission direction.

- **Reference Object.** This method uses a scene object's Z-axis as the target direction for particle emission.

Special Particle Maps

There are two special maps designed specifically for use with particles:

- **Particle Age.** This has three slots for maps or colors that can be blended based on particle age. It varies from particle to particle and is applied relatively.
- **Particle Mblur.** This changes the color of the leading and trailing edges of moving particles. It is most commonly used in the Opacity slot of the material, but can also be used in other slots.

Space Warps for Particles

There are lots of space warps that are used as fields to affect particle motion. They can be applied individually or globally—for example, a single space warp can affect several particle systems at once. Space warps for particles are divided into two groups: Forces and Deflectors. Forces affect the motion of particles during their lifetime and Deflectors work as collision objects for particles.

Forces

The Forces group of particle space warps includes the following:

- **Motor.** This is a dynamic force used to impart a torque effect on particles.
- **Push.** This is a simple repulsive force causing particles to push away along the space warp's direction.
- **Vortex.** This is new to 3ds max 4 and causes a special rotational-well effect similar to the vortex of a tornado.
- **Drag.** This is a useful motion dampener that causes particles to slow down a given percentage over time. This dampening effect can be isolated along certain axes.
- **Path Follow.** This forces particle motion to conform to a shape object in the scene.
- **PBomb.** This is another repulsive force but varies in shape. A repulsive force can be applied spherically, cylindrically, or from a plane.

- **Displace.** This is another useful space warp that uses an image or map to cause motion perturbation. It modulates the amplitude of the forces based on grayscale values, with white as full force and black as no force. The extents of the force depend on the shape of the gizmo being used.
- **Gravity.** This is a common variable force that uses Newtonian physics to cause an attraction or repulsion spherically or planar with acceleration.
- **Wind.** This is similar to gravity and works in both Spherical and Planar mode, but adds perturbation controls in the form of turbulence.

Deflectors

There are three basic categories of Deflectors, differentiated by their shape. All three categories include planar, spherical, and universal (object-based) shapes:

- **Deflector.** This is a standard Deflector that has bounce and friction controls, including Deflector, SDeflector, and UDeflector.
- **DynaFlect.** This is a special type of Deflector used in dynamics simulations. DynaFlect Deflectors have the added capability of enabling the user to include particle systems in dynamics simulations. They add physics-based parameters, such as mass, and include PDynaFlect, SDynaFlect, and UDynaFlect.
- **OmniFlect.** This is a variation of Deflectors that adds the capability of both reflecting and refracting particles. OmniFlect Deflectors include POmniFlect, SOmniFlect, and UOmniFlect.

The Project

Normally, particle systems are used in a subset of animation techniques called *effects animation*. With the exception of effects sequences, they are usually created later in the production pipeline when primary object animation has been completed. In the case of the ongoing project within this book they are being added when the camera has been tracked, the scene stand-in geometry has been matched, and the character and objects have been modeled and animated.

Effects animators are specialists similar to the way character animators are, focusing most of their attention on a complex aspect of animation as a whole. Effects animators tend to differ from character animators in their emphasis and approach to the technique of animation. Character animation focuses heavily on communication through motion and is hands-on, and effects animation tends to be more

technical in nature. One isn't necessarily more difficult than the other, but their disciplines are different. There is definitely more art in character animation and more science in the production of effects animation. Thus many character animators are less package-specific, and effects animators are often married to their packages because they focus so strongly on their implementation of procedural tools.

The goals for effects animation in the project in this chapter are to create an exploding confetti effect when the box top blows off and the Jester character pops out. These will be instanced geometry particles that linger in the air and slowly fall to the ground, spinning along the way. Another effect will be a trailing pixie dust effect coming off the feet of the character as it leaps out of the box. This also will use space warps to help control the motion of the particles. All the effects will be tied together using Parameter Wiring and Custom Attributes to more easily adjust the effects in the scene.

Exercise 21.1 Setting Units

One of the first things to do with your scene when setting it up for work with particle effects is to adjust the units method. *Units* are the method used to represent values in parameters. This is important when modeling to scale, because you can set the values to real-world units such as decimal feet or meters. However, such units have no relevance to particle values, such as velocity and birth rate. You need to set the units of your scene to generic, which uses standard integer and floating-point values with no real correlation to real-world units.

1. Select File/Customize/Units Setup. At the bottom of the resulting dialog box, set the radio button to Generic Units, as seen in Figure 21.1.

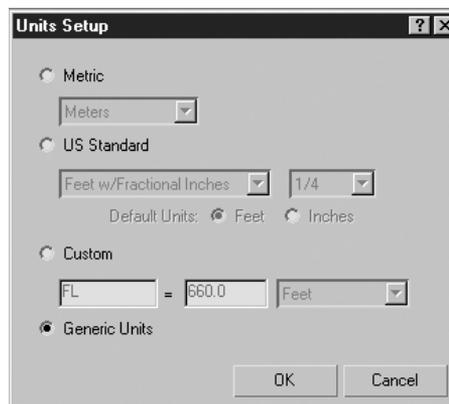


Figure 21.1 Select Generic Units in the dialog box.

Next you use PArray to create a trail of particles from the feet of the character.

2. In the Create panel, choose Particle Systems/PArray.
3. Click and drag anywhere in the Top viewport to create a PArray icon, as in Figure 21.2.

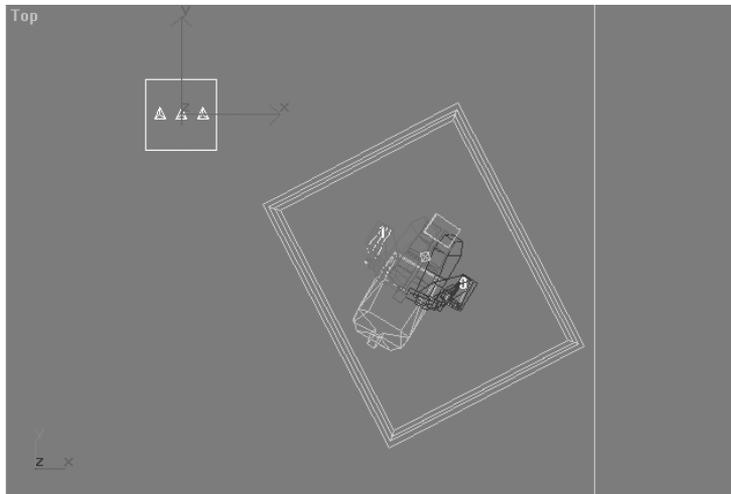


Figure 21.2 Add a PArray icon to the scene in the Top viewport.

There are a few PArray parameters you need to set up.

4. With the PArray selected, go to the Modify panel.
5. In the Basic Parameters rollout, set the Object Based Emitter by clicking the Pick Object button and choosing Bip01 R Foot from the list.
This sets the emitter to the biped's right foot.
6. Set the Viewport Display section to Dots.
This makes it easier to view the particles, because they are going to be represented by tiny pixie dust points.
7. Set the percentage of particles to **100%** so you can see the entire quantity for now.
8. In the Particle Generation rollout, set Speed to **0.1**.
The scale of this scene is pretty small, so the default value of 10 would be much too fast.
9. Set Variation to **25%**.
This varies the speed of particle emission by 25%.

10. Set Emit Start to 60.

This is the frame in which particles will first be emitted, when the character jumps out of the box.

11. Set Emit Stop to 96.

This is the frame in which particles will stop emitting and is the point when the character first reaches the surface of the table.

12. Set Display Until to 204.

This is the frame in which the particles disappear. This might seem pointless, because the particle emission ends at frame 96, but it's good practice to adjust this value from its default of 100 to at least one frame past the end of your animation. It's a common mistake to ignore this parameter, which results in all the particles disappearing at frame 100.

13. Set Life to 90.

This gives the particles a lifetime of 90 frames.

14. Set Variation to 5.

Unlike the other variation parameters, this one is in frames, not percentages. This causes the particle life value to vary plus or minus 5 frames. This reduces the uniformity of the effect by adding some randomness.

15. In the Particle Type rollout, set the standard Particles to Constant.

This makes the particle geometry constantly scaled circular points that always face the camera.

16. In the Object Motion Inheritance rollout, set Multiplier to 0.2.

This causes the particles to receive 20% of the motion of the emitter—in this case the biped foot. This is important because in the real world when an object is ejected from a moving object, it begins its motion at the same speed as its parent.

17. Scrub the animation by clicking and dragging the time slider.

You'll see a trail of particles emitting from the right foot of the biped object. Because of the motion inheritance, the particles aren't stationary but actually move with the foot as they're emitted. See Figure 21.3.

There is one problem, however. The particles are emitted too fast. There also needs to be a drag and gravity effect applied to them. This is done through space warps, which you tackle in the next exercise.

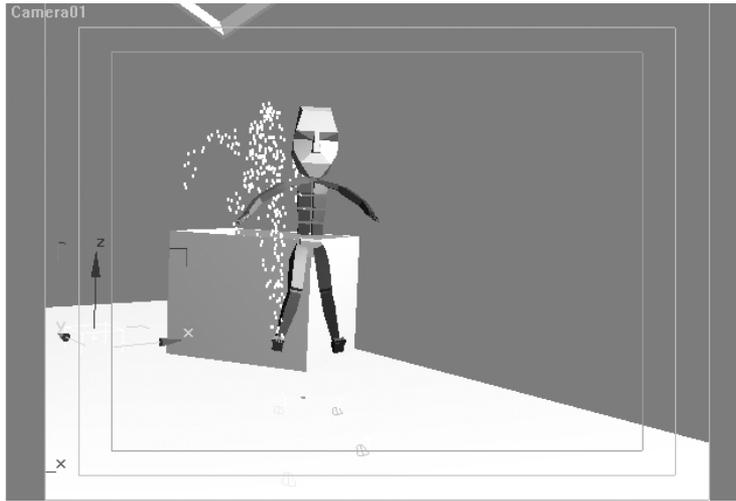


Figure 21.3 PArray particles trail from the right foot of the biped.

Exercise 21.2 Adding a Drag Effect

In this exercise, you add a Drag space warp to cause a dampening effect on the particles' velocity. *Space warps* are nonrendering objects that act as fields in the scene and are accessed through the Create panel like any other object.

1. Go to the Create panel and click the Space Warps button.
2. From the Forces drop-down list, choose Drag.
3. Click and drag a Drag icon into the Top viewport.
4. With the Drag space warp selected, go to the Modify panel.

All the Drag parameters appear in the Modify panel. There are many for such a simple effect. The new Drag space warp offers much flexibility, including nonlinear and localized dampening, so the effect doesn't have to be uniform (on all axes) or affect the particles globally across the entire scene.

You need to set a few parameters for this Drag space warp.

5. Set Time Off to 205.

Normally this is set to the default value of frame 100, which causes the Drag effect to end at frame 100, in the middle of the animation. It's generally good practice to change Time Off to match the end frame of your animation, which causes the Drag effect to continue until the end of the animation and not turn off inappropriately.

6. Set the Linear Dampening X-axis and Y-axis to 2.0.

This sets an equal amount of drag force along the X-axis and Y-axis.

7. Set the Linear Dampening Z-axis to **10.0**.

This increases the Drag effect along the Z-axis, which prevents the particles from moving too much vertically as they are emitted. They also will be affected laterally in the X-axis and Y-axis, however, but much less. This gives them some freedom to wander. See Figure 21.4.

The particle system is now bound to the space warp. If you go to the Modify panel with the Parray01 selected, you'll notice an entry in Stack View that shows a Drag Binding.

8. Play the animation by either scrubbing the time slider or clicking the Play button.

The particles are emitted from one of the feet of the biped, but slow down a bit and linger. There's still one more step to control the motion—adding turbulence—which you'll do in the next exercise. By using the Turbulence controls of a Wind space warp you can perturb the motion of the particles slightly as they linger in the air. This breaks up the uniformity of the effect, making it more realistic and natural.

Exercise 21.3 Adding a Wind Space Warp

You create a Wind space warp the same way you create a Drag warp. In fact, the Wind space warp is similar to the Gravity space warp because it has a directional icon showing the vector of the effect.

1. Go to the Create panel, choose Space Warps/Forces, and click Wind.
2. Click and drag in the Top viewport to create a Wind space warp.
3. Bind the particle system to the Wind space warp by selecting the Parray01 particle system.
4. Activate the Bind to Space Warp button with a single button-click.
5. Click and drag from the Parray01 particle system to the Wind space warp icon and release the mouse button.
6. Verify that the space warp is bound by checking Stack View for the Wind binding. Use Figure 21.5 as a reference.



Figure 21.4
Drag space warp parameters increase flexibility.

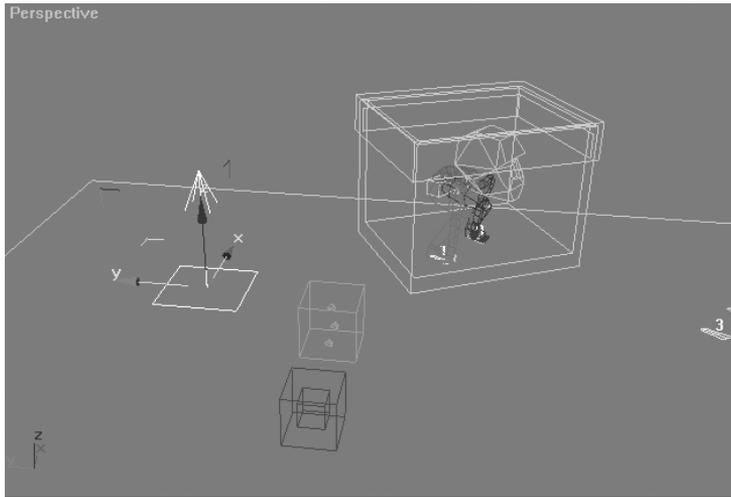


Figure 21.5 Wind space warp is added to the scene.

By default, Wind has a Strength value of 1.0. This is the amount of force in the direction of the Wind space warp icon's arrow—in this case, upward. If you scrub the animation at this point, you'll notice that too much force is applied to the particles and they are heavily displaced upward. If you select the Wind space warp, go to the Modify panel and look at the first two Turbulence parameters. Turbulence and Frequency are zero, which means that there is no perturbing effect on the particles. To create turbulence without forcing the particles in any given direction, you have to turn off the Strength value.

7. To turn off the Strength value, select the Wind space warp.
8. Go to the Modify panel and change the Strength value to 0.

Now if you take a look at the animation, the particles don't launch upward in the direction of the Wind icon. No force is being applied in the Wind's direction. However, you still need to make adjustments to the Turbulence parameters to achieve the proper effect.

9. Adjust the Turbulence value to 0.02.
10. Change the Frequency value to 0.15.
11. Set the Scale value to 0.2.

With those numbers plugged in, you should see some extra movement in the particle motion. Play with these numbers to experiment with the motion, but don't wander too far from these values because the effect can quickly become unrecognizable. At this point you leave the pixie dust effect for a bit and move on to a confetti effect.

**Note**

These might seem like arbitrary numbers, but I consider them default parameters for most scenes. These numbers are a good starting point for integrating turbulence into a scene. Many people often make the mistake of using higher numbers, such as 2.5 or 10. Unfortunately, this doesn't work because so much depends on the scale of your scene.

Keep the following in mind:

Turbulence controls the speed at which particles waver and curl or are perturbed from their normal paths. Lower amounts cause gentle motion, and higher values naturally cause excited motion.

Frequency is directly related to Turbulence. Frequency controls the amount of randomness in the motion over time. Higher numbers cause more variation.

Scale is probably the most important value to set correctly, because without matching this value to the scale of your scene, the effect is unnoticeable. The number values for this parameter might be misleading. Very low numbers—0.01, for example—cause large turbulent motion, and higher numbers cause tight motion. Turbulence is really just a fractal noise field applied to the particles. The Scale value controls the scale of the noise and therefore the visibility of the motion.

Exercise 21.4 Creating a Confetti Effect

In this exercise, you use a SuperSpray particle system emitting particles as instanced geometry to create a confetti effect. By using instanced geometry you'll be able to use a custom object as particle geometry. You'll also add some space warps to control the motion of the particles, and finally add some Custom Attributes and Wire Parameters to better control and automate both effects.

To start the confetti effect, you need to choose a particle system. SuperSpray is ideal in this case because it has similar parameters to the PArray particle system you used previously, but also has the capability of emitting particles from a single point outward in a controlled direction.

1. Go to the Create panel, choose Particle Systems, and click SuperSpray.
2. Click and drag in the Top viewport over the box01 object to create a SuperSpray icon, as seen in Figure 21.6.

**Note**

The SuperSpray icon is slightly different from the PArray particle system icon. SuperSpray emits particles in a direction determined by the icon, not by a geometric object in the scene. In this case, the location of the icon and the direction you point it in determine how particles are emitted. By creating the SuperSpray particle system in the Top viewport, you automatically create it pointing upward along the Z-axis. Notice there is an arrow built into the icon.

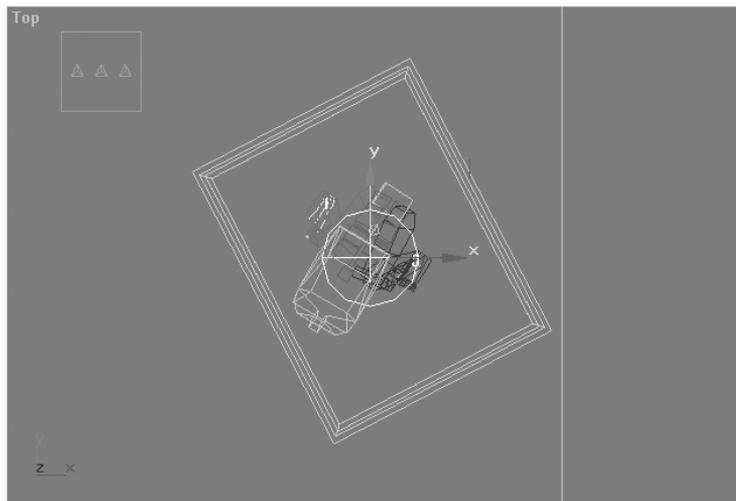


Figure 21.6 The SuperSpray particle system is located inside the gift box with the biped.

Before you go any further, there are a few parameters you need to adjust.

3. In the Basic Parameters rollout, set Percentage of Particles to **100%**.
This causes every particle emitted internally to be displayed in the viewports.
4. In the Particle Generation rollout, set Use Total to **100** by switching the radio button to activate this parameter and adjusting the total number of particles to be emitted to 100 particles.
5. In the Particle Motion subsection, set Speed and Variation to **2** units and **25%**, respectively.
These two values are related and therefore mentioned in the same space.
6. In the Particle Timing subsection, set Emit Start to **60**.
This defines the starting frame for particle emission.
7. Set Emit Stop to **65**.
This causes all particle emission to stop at frame 65. With these two start and stop values set accordingly, the total number of frames in which particles will be emitted is 5.
8. Set Display Until to **204**.
This should normally be a number beyond the final frame of your animation.
9. Set Life and Variation to **100** and **10**, respectively.
This defines the number of frames that particles will be “alive,” give or take 10 frames. The Variation parameter is different from most others because it is defined in frames, not percentage. In this case when a particle is born its Life value is determined to be between 90 and 110.

If you take a look at the animation of the SuperSpray particles you see that they emit upward in a tight stream. This is a matter of adjusting some axis and spreading controls. You'll be able to better visualize these changes if you move the time slider to a frame where the particles are visible.

10. Go to frame 65 and select the SuperSpray particle system.
11. Go to the Modify panel and change the Off Axis Spread value to **80** and the Off Plane Spread value to **180**, as seen in Figure 21.7.

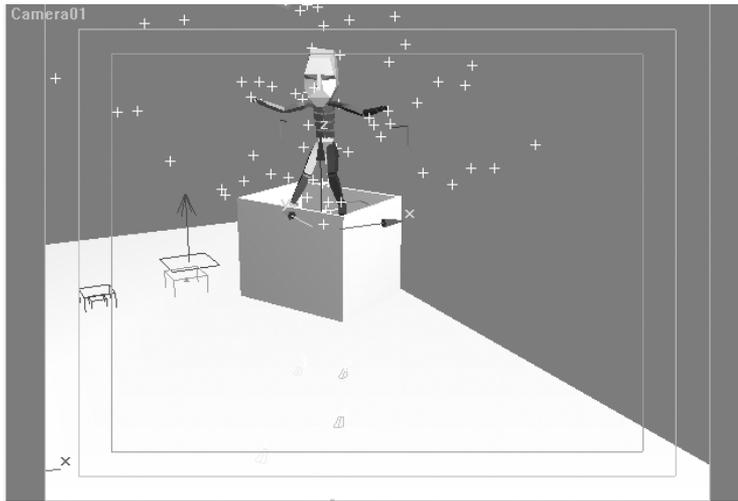


Figure 21.7 Adjust the Off Axis Spread and Off Plane Spread controls to create a more conical emission.

As you adjust these two values, notice what they do to the angle of the particle emission. When you adjust the first parameter, the emission is in the form of a flat fan. However, as soon as you adjust the second parameter, the shape of the spreading fan becomes conical. These are two powerful controls for shaping particle emission.

The next step is to assign instanced geometry to the SuperSpray particles. Recall that instanced geometry is a method of referencing an object in the scene to use as particle geometry. In this case, you need to make a confetti strip object to use as the instanced geometry in the particle system.

To create the confetti strip, you use a Plane primitive and apply a couple of modifiers to shape it. The modifiers will come in handy, because they enable you to edit the object's shape without destroying it.

12. Go to the Create panel, choose Standard Primitives, and click the Plane button.

- 13.** In the Top viewport, click and drag to form a plane primitive in the scene. The initial size of the plane primitive is irrelevant, because you'll adjust its parameters in the Modify panel.
- 14.** With the plane primitive selected, go to the Modify panel.
- 15.** Change the Length value to **2.0**, change the Width value to **0.5**, adjust the Length Segs value to **13**, and rename the plane primitive **ConfettiStrip**.
If you zoom in on the object, you'll see that it's a long, thin, single-sided object made up of evenly spaced polygons. The reason for adding more Length Segs past the default value of 4 is to make the polygons even. This becomes necessary when you're going to bend and twist the object, which you'll do next.
- 16.** With the new ConfettiStrip object selected, go to the Modify panel.
- 17.** Click Modifier List to open the drop-down list and choose Bend.
This adds a Bend modifier to the object, which enables you to warp the object on a specific axis a set amount of degrees. The motivation here is to skew the ConfettiStrip into a twisted piece of paper.
In the Bend parameters rollout, you make a few changes.
- 18.** Set Angle to **-60**.
This bends the modifier gizmo -60 degrees.
- 19.** Set Bend Axis to X.
This restricts the bending effect to the X-axis of the gizmo.
If you zoom in on the ConfettiStrip object in one of the side viewports, you can see the bending effect the modifier has on the object.
Next you add a Twist modifier, which is similar to a Bend modifier but has a more asymmetrical effect.
- 20.** With the ConfettiStrip object selected in the Modify panel, click Modifier List to open the drop-down list and select Twist from the list.
This adds a second modifier onto the object's stack, visible in Stack View. As with the Bend modifier, there are two parameters you need to adjust.
- 21.** Set Angle to **50**.
This increases the twist effect by 20 degrees.
- 22.** Set Twist Axis to Y.
This isolates the effect to the Y-axis of the modifier's gizmo, as seen in Figure 21.8.
With these steps complete, you should have a long strip of polygons twisted and bent into a confetti shape. The next step is to tell the SuperSpray particle system to use this newly created shape as instanced geometry.

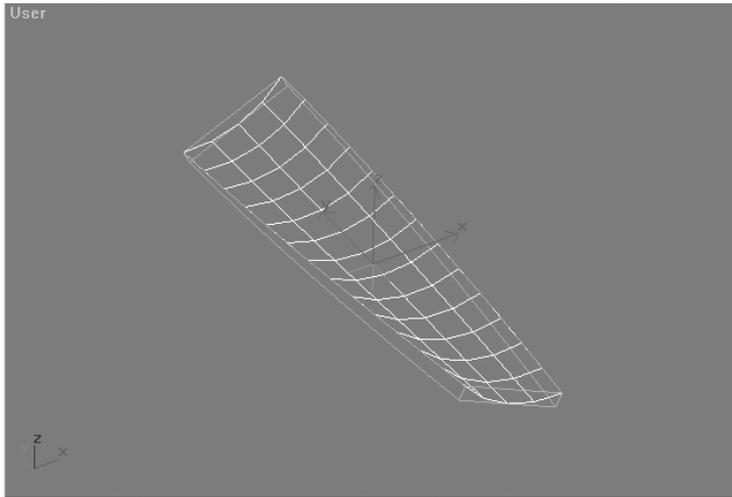


Figure 21.8 Apply Bend and Twist modifiers to ConfettiStrip.

23. Select the SuperSpray particle system. Go to the Modify panel to access its parameters.
24. In the Particle Type rollout, change the radio button to Instanced Geometry.
25. Click the Pick Object button below in the Instancing Parameters and choose the ConfettiStrip object.

In your viewport, this doesn't make any immediate changes because the particles are still being displayed in their default manner as constant-sized Ticks. You can set the display option to show the actual geometry of the particles, though.

26. Go to the top rollout of the SuperSpray particle system and set the Viewport Display radio button to Mesh.

Now you should see the ConfettiStrip object being instanced to each particle of the SuperSpray particle system. If your viewport display feedback is too slow, you can opt to set Viewport Display to Bbox. This reduces the detail of the instanced geometry displayed by showing only its bounding box. One problem does exist, though. The original ConfettiStrip object is still present in the scene and needs to be hidden.

27. To hide the original ConfettiStrip object, select it, go to the Display panel, and click Hide Selected, as shown in Figure 21.9.

This doesn't delete the object, but hides it from view because you are not going to actually use it for anything other than reference for the SuperSpray particles.

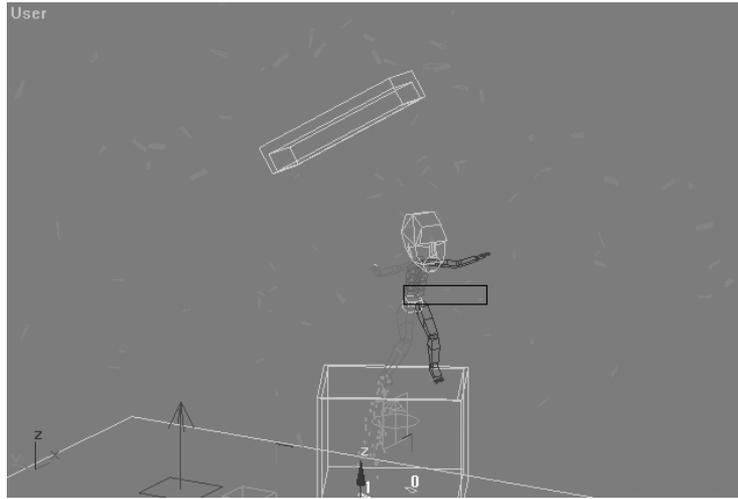


Figure 21.9 SuperSpray uses the ConfettiStrip object as instanced geometry.

Using instanced geometry is a useful way to extend the geometric representations of particles in 3ds max. You can use polygons, patches, and even NURBS as instanced geometry. Unfortunately, there are a few drawbacks. In particular you can instance only one particular object and its hierarchy, not a group of objects. For example, if you create geometry for coins—such as dimes, nickels, and quarters—you have to create a separate particle system to instance each type of object.

Exercise 21.5 Adding Drag, Wind, and Gravity

You now have the basic setup for the confetti effect. One of the problems at the moment is the unrealistic motion of the particles. They should explode out, slow down, and fall to the floor like flower petals. As you did previously with the PArray particle system, you can use space warps to control the motion of the SuperSpray particles. Drag and Wind space warps are present in the scene, but the only one you're going to reuse is the Wind space warp for its turbulence effect. Space warps can be used with multiple objects, and any changes in their parameters will affect those objects bound to it.

First you need to add a second Drag space warp with stronger, more uniform force.

1. Select the Drag01 space warp in your scene.
2. Switch to the Move tool by right-clicking the space warp and choosing Move from the quad menu.
3. Hold down the Shift key, drag the Drag space warp over so that it is separated from the original, and release the mouse.

4. In the resulting Clone Options dialog box, choose OK. Figure 21.10 shows the result.

You have just created a clone of your original Drag space warp. They have the same values in all parameters, with the exception of their names. This new Drag space warp needs to be stronger to slow down the exploding SuperSpray particles.

5. With the new Drag02 space warp selected, go to the Modify panel.
6. Change all three Linear Dampening X-, Y-, and Z-axis values to 5%.
7. Select the SuperSpray particle system and bind it to the new Drag02 space warp using the Bind to Space Warp button.

If you take a look at the animation now, at frame 60 the SuperSpray particles explode outward in a hemispherical pattern and quickly slow down so that they create a sort of confetti effect.

Two problems are obvious. First, the particles should look like little strips of paper, but they don't spin. Second, they don't fall downward toward the floor, essentially defying gravity. Both of these problems can easily be addressed.

First, rotation is merely a matter of adjusting a couple of parameters in the particles. Second, Gravity, like Wind and Drag, is a space warp that you add to the scene.

8. Select the SuperSpray particle system and go to the Modify panel.
9. In the Rotation and Collision rollout, set the Spin Time to 10 frames.
10. Set the Phase to 30.0 degrees.

Each of these parameters has a Variation parameter. This adds randomness to the values and breaks up the uniformity of the effect—something you should always strive for. It's also useful to use the Wind space warp's Turbulence values that are already in the scene to perturb the motion of the confetti particles.

11. Set both Spin Time Variation and Phase Variation values to 25.0. Figure 21.11 shows the result.
12. Activate the Bind to Space Warp button and bind the SuperSpray particle system to the Wind01 space warp already in the scene.

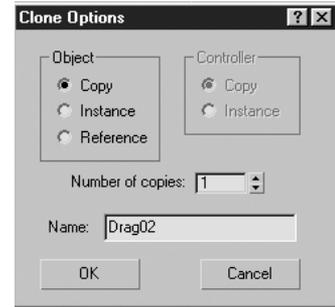


Figure 21.10

The Clone Options dialog box appears when you drag the Drag space warp while holding down the Shift key.

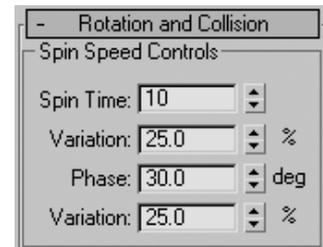


Figure 21.11

Set the Rotation and Collision parameters.

The only other force missing from this effect is Gravity, just another simple space warp to add to the system. Gravity works similarly to Wind because it is a planar directional effect by default. It can also be a spherical effect, as can Wind, but in this case the default setting works best to mimic the gravity of the sample scene.

13. In the Create panel, choose Space Warps/Forces and click Gravity.
14. In the Top viewport, click and drag a Gravity space warp icon into the scene.
15. Select the SuperSpray particle system and bind it to the new Gravity space warp. Figure 21.12 shows the result.

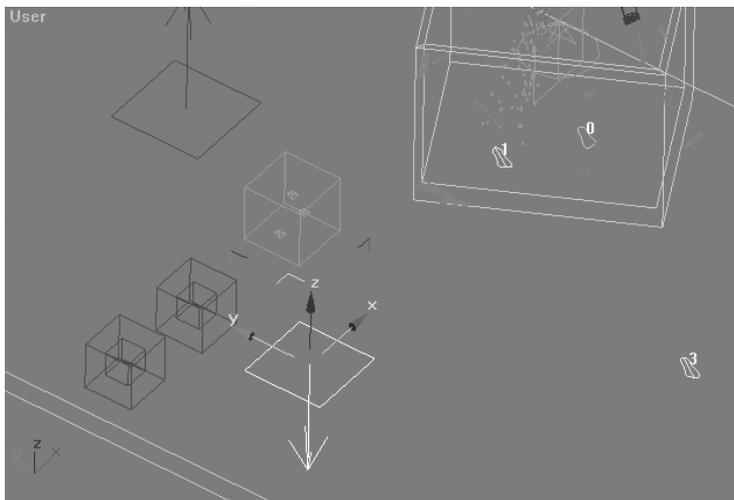


Figure 21.12 Add a Gravity space warp to the scene.

Now the SuperSpray particles explode outward, but fall downward too fast. You can fix this by reducing the strength of the Gravity effect. Instead of doing this immediately by modifying the value, do some parameter consolidation with Custom Attributes.

Exercise 21.6 Adding Custom Parameters Using Custom Attributes

Custom Attributes are a new feature in 3ds max 4 that enable you to add custom parameters to objects. Using Parameter Wiring, you can connect Custom Attributes to parameters in your scene, using them to drive animation or, in this case, to consolidate several parameters from different objects and store them in one easily accessible rollout. It might sound complicated, but it's actually pretty simple. Just follow the steps in this exercise.

You create a dummy object that will not render or participate in any way in the scene, other than to store several parameters that you can use to manipulate the scene.

1. In the Create panel, choose Helpers/Standard and click the Dummy button.
2. In the Top viewport, away from all the clutter in the scene, click and drag to create a Dummy object.
3. With the Dummy object selected, go to the Modify panel.

The Dummy object has no parameters. You can use Custom Attributes to add parameters here that you define. Custom Attributes is accessed in the Animation menu at the top of the screen.

4. Select Animation/Add Custom Attribute.

This causes a large floater to appear with lots of controls. This is sort of like a custom parameter construction set. First you add a Gravity Strength control.

In the Add Parameter floater, the default parameter type is Float and the UI type is Spinner. Use these defaults.

5. Change the Name field to **Gravity**.
6. Open the Float UI rollout and set the Width value to **100.0**.
7. Set Range From to **-1** and Range To to **1.0**.
8. In the Finish rollout, click the Add button. Figure 21.13 shows the result.
A new parameter called Gravity appears in the dummy's Custom Attributes rollout. If you adjust this parameter, you'll see that it doesn't really do anything at the moment. The spinner field is 100 units wide, and the parameter's value range is from **-1** to **1.0**. You need to connect it to the Gravity space warp's Strength value by using Parameter Wiring.
9. Zoom out the Top viewport so you can clearly see both the Dummy object and the Gravity space warp.
10. Right-click on the Dummy object and select Wire Parameters from the Quad menu.
11. In the resulting pop-up menu, choose (in this order) Object(Dummy), Custom_Attributes, and Gravity.
12. Now click on the Gravity icon. Notice a line drawn from the dummy.
13. In the pop-up menu choose (in this order) Object(Gravity), and Strength. Figure 21.14 shows the result.

This connects these two floating-point values internally via MAXScript. The resulting dialog box is the Wire Parameters dialog box where multiple wiring can be edited.

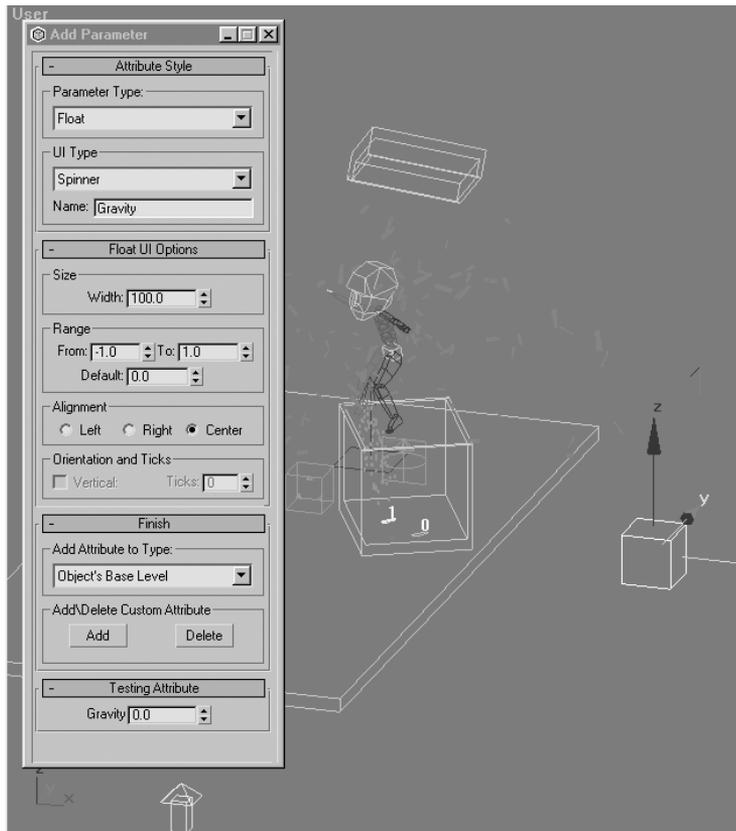


Figure 21.13 Set the controls in the Add Parameter dialog box.

The Gravity track is highlighted on the Dummy object on the left side of the box, and the right side shows the Strength track of the Gravity space warp. This identifies the two tracks that you are wiring together. Parameters can be wired in directions so that adjusting one track doesn't necessarily affect the other. Tracks can also be bidirectional, meaning that changing the values of either affects the opposite track. In the two lower boxes, there are text boxes with variables for the track values. This is actually an expression field and can be used to adjust how a track is interpreted. In the case of Custom Attributes that you're setting up, you want them to be bidirectional.

14. In the Parameter Wiring dialog box, click the Two-Way connection button. It's located at the top and looks like two opposing arrows.
15. Click the Connect button to create the connection. See Figure 21.15 for the result.

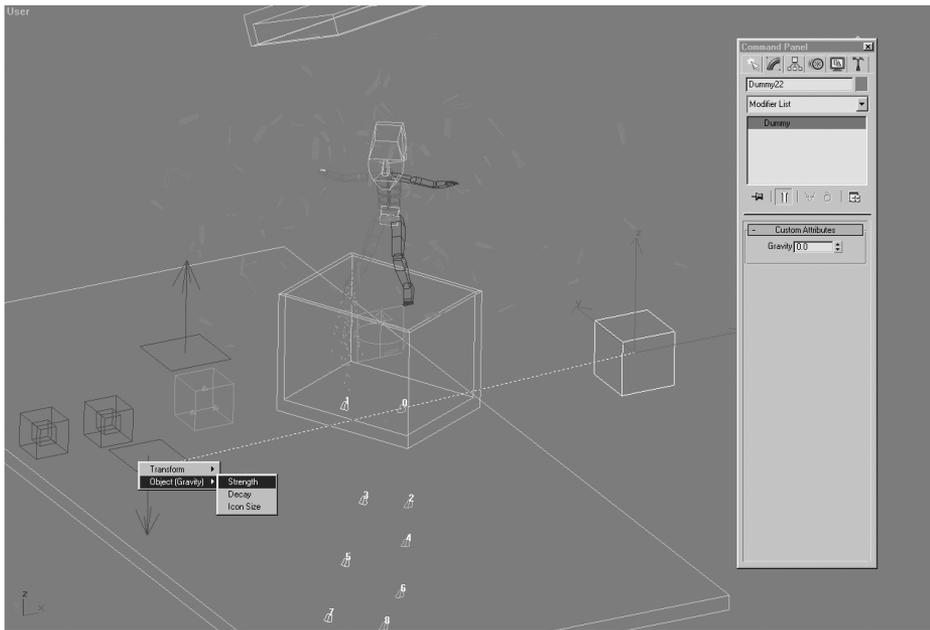


Figure 21.14 Wire the Custom Attribute to the Gravity's Strength value.

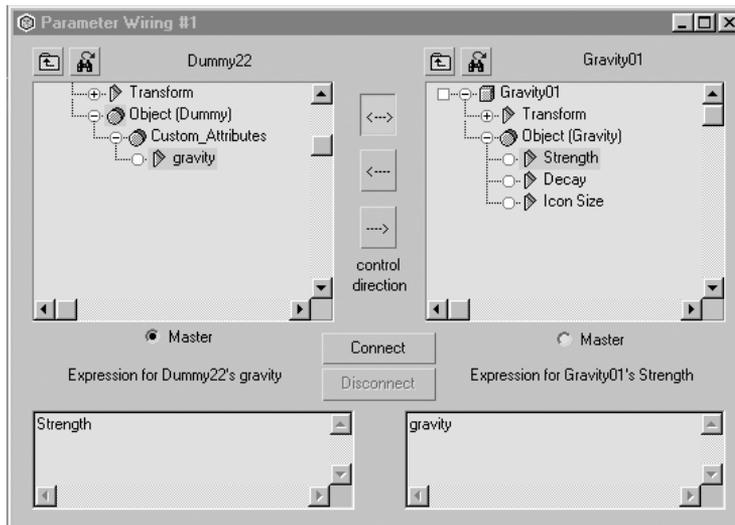


Figure 21.15 Complete the Parameter Wiring.

Now you've connected the Custom Attribute Gravity to the Strength value on the Gravity space warp. If you change the value of the Custom Attribute, the Gravity space warp's Strength value adjusts accordingly. Custom Attributes are fairly limitless, and using this method of storing them on a common object enables you to organize many parameters from several dissimilar objects into one consolidated interface.

The animation is nearly complete, but there still are a few tasks to do to complete the scene. Use the Custom Attribute you created to adjust the strength of the Gravity to a lesser amount more consistent with the scene.

16. Select the Dummy object and change the Gravity Custom Attribute value to **0.02**.

This will work with the Drag space warp to cause the confetti particles to linger in the air and slowly float downward. The confetti particles scale up quickly and then scale down as they die. This is a function of the Grow For and Fade For values in SuperSpray. The default parameters of 10 cause the particles to scale up from zero to their full size in 10 frames and fade out to nothing 10 frames before each dies.

Parameter Wiring and Custom Attributes are a huge windfall for 3ds max users. They extend the procedural tools to a new level by enabling you to easily set up and control complex interactions between parameters of dissimilar objects. With Custom Attributes, the user is free to add specialized parameters to objects or, as in this example, consolidate parameters in the scene to one object.

Exercise 21.7 Adding Materials and Effects

The last big step in this process is to assign special materials and effects to the particles. First, you use a Multi Sub-Object material to randomly assign different colors to different strips of confetti. Then you use the Render Effect Lens Effects to add a glow effect to the pixie dust particles. Finally, you adjust the properties of the particles to better integrate them into the scene.

1. Open the Material Editor and choose an empty slot, or make one by selecting an existing slot and clicking the Delete button, which looks like an X below the material slots).

If you are deleting a slot containing a material that exists in your scene, you are asked whether you want to delete the material from the scene (bad) or just remove it from this particular slot (good).

2. Create a Standard Material by clicking the Material Type button and choosing Standard from the list.
3. Change the Diffuse color swatch to a bright color, such as red.

4. Adjust the self-illumination value to 25.
5. Click the 2-Sided check box.
6. Rename the material **Confetti** and drag it onto the SuperSpray particle system to assign it.

**Note**

By adding a fraction of self-illumination, you fake translucency in the confetti strips, an effect common with thin paper. By clicking the 2-Sided check box, you define this material as an automatically two-sided one, regardless of the geometry it's assigned to. This is important because the ConfettiStrip object you created was initially created with a Plane primitive that has polygons facing in only one direction. This essentially makes it one-sided, and viewing from directly underneath the plane causes it to be invisible. By assigning a two-sided material to the object, you're automatically causing the object to be two-sided, overcoming the shortcoming of plane without adding geometry.

What if you want the confetti to be several different colors? With the tedious method, you create multiple copies of the confetti particle system. Controlling these separate systems could be a hassle. With the better method, you assign a Multi Sub-Object material, and the particles will cycle through the maps in the list on a frame-by-frame basis, assigning different materials to particles at the given frame in the cycle. This doesn't work when particles are emitted over the course of a single frame. Luckily, in this scene the particles emit over the course of five frames.

7. To change the Confetti material to a Multi Sub-Object material, select the Confetti material in the Material Editor.
8. Click the Material button (which should read Standard at the moment) and choose Multi Sub-Object from the list.
9. In the resulting Replace Material dialog box choose Keep Old Material as Sub-Material.

This converts the Standard material into a Multi Sub-Object material with the default 10 entries. In the case of the confetti effect, you need only 5 entries.

10. Click the material's Delete button five times to remove materials 6 through 10 on the list.

You've already defined one material with all its special parameters (the red one), so instead of going in and matching all the settings with the new materials, copy the red Confetti material slot into all the other slots. From there you can adjust the color for each without having to go into each material slot.

11. Click and drag the top material (the red Confetti material) slot to the next one below.

12. Set the radio button in the resulting Instance (Copy) Material dialog box to copy and press OK.
13. Repeat this until all five slots look the same.
In this sequence, you've copied the red Confetti material parameters to all the other slots in the Multi Sub-Object material. Now all you need to do is adjust the colors of the rest.
14. Click on the color swatch to the right of each submaterial and change it to a different color (try to stick to bright confetti-style colors, such as pink, aqua, and yellow). Figure 21.16 shows the result.

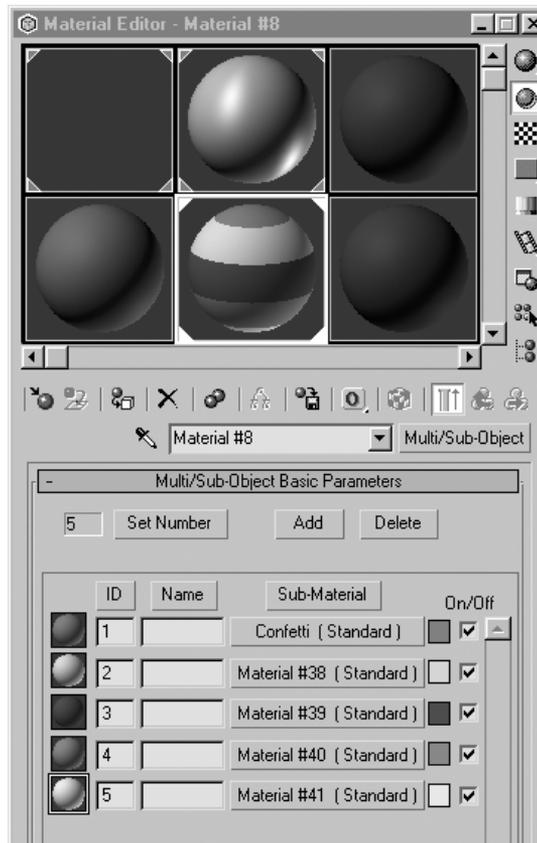


Figure 21.16 Store several materials in a Multi Sub-Object material.

When you look in the viewport, you won't notice any change in the color of the confetti particles, because, by default, the SuperSpray particles are getting their material from the instanced geometry and not the icon. This is easily changed in the SuperSpray parameters.

15. Select the SuperSpray particle system.
16. Go to the Modify panel. At the bottom of the Particle Type rollout, switch the Get Material From radio button to Icon.
17. Play the animation.

The colors of the confetti particles vary according to the list of materials stored in the Multi Sub-Object material. Figure 21.17 shows the result.

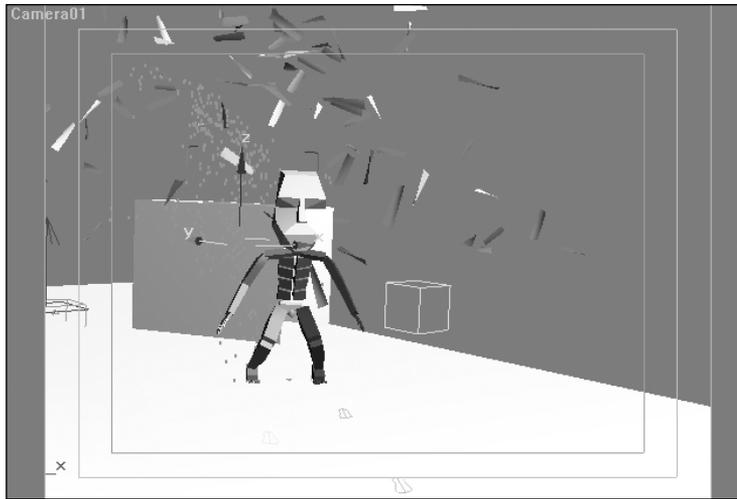


Figure 21.17 The confetti particles use a Multi Sub-Object material.

The next step is to assign a Glow effect to the PArray particles that are serving as pixie dust. At the moment they are rendering as tiny constant, or circular, geometry that always faces the camera. They have a default material attached to them that must be changed to make them more colorful. Luckily, you can use the same Multi Sub-Object material you created for the confetti particles.

18. Go back to the Material Editor and drag the Confetti material onto the PArray icon.



Note

There are a couple of ways to assign materials to objects, but this is one of the more direct. You can also use the Assign Material to Selection button in the Material Editor, but that requires a selection. The drag-and-drop method is quicker because it doesn't require any selections.

The viewport display of the PArray particles doesn't show multiple colors. Don't worry—currently the PArray particles are being displayed only as dots and not geometry. To verify that the assignment of multiple materials is working, try rendering out a frame to the virtual frame buffer.

Exercise 21.8 Using Render Effects

Render Effects are a post-process filter tool that was added in 3ds max r3. There are several kinds of Render Effects, but you work with an optical effects toolkit called Lens Effects in this exercise.

Lens Effects has tools for creating glows, highlights, sparkles, secondary reflections, and other optical anomalies via post-process. In the past, these kinds of filters resided only in Video Post and required much more time to work with. Moving Lens Effects to a Render Effect filter brings speed and flexibility to the workflow.

Render Effects can be added and edited with a single test render. Normally you have to re-render your scene to see any changes you make with a filter, but in the case of Render Effects, the filter is passed through a valid scene enumeration that can be stored in memory and applied to filter changes on-the-fly. This enables you to do much of your filter editing in near-real time.

In this exercise, you need a tiny subset of the Lens Effects toolkit: Glow.

1. From the top menu, choose Rendering/Effects.
2. Click the Add button and choose Lens Effects from the list.
3. Select the newly added Lens Effects entry.

A few new rollups are added. One contains two windows with elements on the left, and one is empty on the right. The empty one is where you assign Lens Effects tools to the filter. The rollout below stores Lens Effects global variables that can be used to apply common parameter changes across several elements.

4. In the window on the left, choose Glow from the list and click the > arrow.
5. Select the Glow text in the right window. Figure 21.18 shows the result.
Notice how a new set of rollouts is added below when you select the Glow element. These are the local controls for that tool. In the next steps, you adjust a series of parameters to create the pixie dust effect for your scene.
6. In the Parameters tab of the Lens Effects Globals rollout, change the Size value to **5.0**.
7. In the Parameters tab of the Glow Element rollout, adjust the size to **3**.
8. Change Use Source Color to **100**.
9. Click the Options tab.
10. In the Apply Element To field, click the Image Centers check box and clear the others.
11. In the Image Sources field, click Object ID and clear any others. Figure 21.19 shows the result.

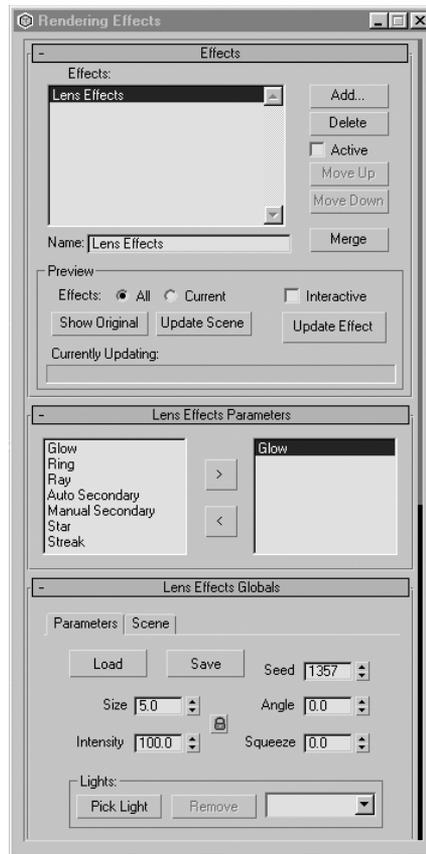


Figure 21.18 Select the Glow text in the Rendering Effects dialog box.

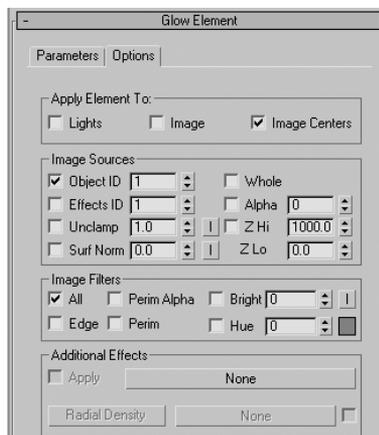


Figure 21.19 Adjust parameters for the Glow element.

**Note**

Here's a breakdown of what you just did. You adjusted the global scale of the Lens Effects filter. By default it's 100, which is too large for this scene. If you had multiple elements, you could use this global size value to adjust the scale of all elements equally at the same time. By setting the Use Source Color value to 100, you're telling the glow filter to glow the particles using each particle's material color 100%. This is a handy feature, though you can explicitly color Glow elements using different parameters in the filter. Finally, by setting Image Centers and Object ID 1, you're telling the filter to apply the Glow effect only to the center of nodes in the scene that carry an Object ID of 1. Currently, you have no objects with an Object ID of 1, so you need to set one.

12. Select the PArray particles and right-click directly over them.
13. Choose Properties from the Quad menu.
14. Set the Object Channel value to 1.
15. Clear the Cast Shadows check box. Figure 21.20 shows the result.

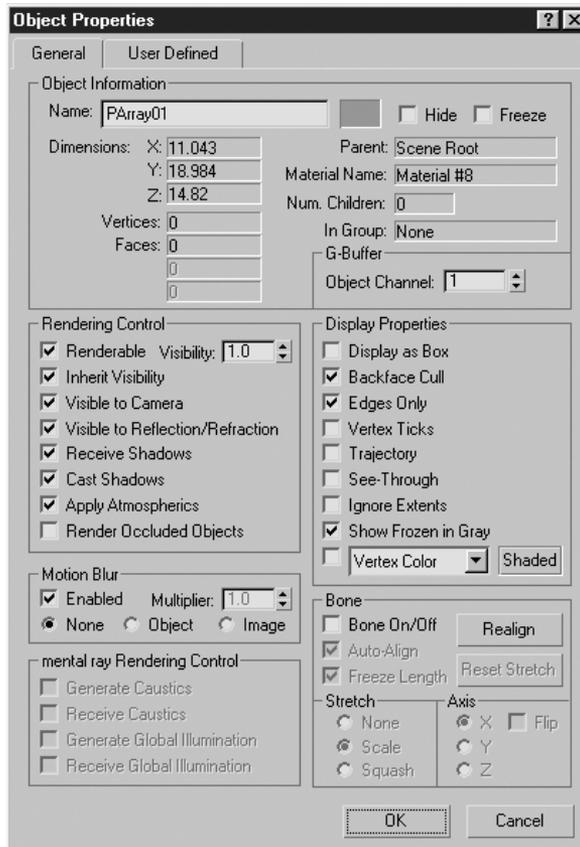


Figure 21.20 Set controls and properties in the PArray Object Properties dialog box.

Now if you render the Camera view, it first renders the entire scene, and then a Render Effects pass is processed. This can sometimes be very fast, but in the case of working with these particles it might be a bit slow.

The Render dialog box shows you the progress of the render effect, as seen in Figure 21.21. This glow effect is driven by Object ID 1 in the scene, so if any other objects in your scene were tagged with that ID they glow as well. You clear the Cast Shadows check box to force the PArray particles to not cast shadows onto other objects in the scene. Normally you'd think that this is important for integration into a live action shot, but in the case of these glowing particles they're appearing as a light source and therefore shouldn't cast any reasonable shadow.



Figure 21.21 The Glow effect is applied to particles.

The last thing you need to do with the PArray particles is to clone them so that you have a second stream that flows from the other foot of the biped. Instead of creating a second PArray particle system from scratch, use the Clone tool, because all the parameters are finalized in the original PArray.

16. Select the PArray particle system.
17. Choose Edit/Clone from the top menu.
18. In the Clone Options dialog box, accept the default name by clicking OK.

This creates an exact duplicate of the previous PArray particle system and—more importantly—its particles have the same path. Why? They are being emitted from the same object and use the same Seed value. You need to adjust these two parameters to complete the effect.

- 19.** With PArray02 selected, go to the Modify panel and change the Object-Based Emitter to Bip01 L Foot.
- 20.** Scroll to the bottom of the Particle Generation rollout and click the New button next to Seed to create a new randomly generated Seed value.

Now if you play the animation you should have two sets of particles being emitted from the feet of the biped object. This concludes this section of the project.

Summary

As you can see from this chapter, particle systems are a useful, though specialized, tool in the 3ds max 4 tool set. They can easily and quickly animate lots of objects together, but they are specialized in their application and not always required in a project. Space warps go hand in hand with particle systems, often being required to integrate a system into a scene or to simply “wrangle” the particles in the direction you want them to go.