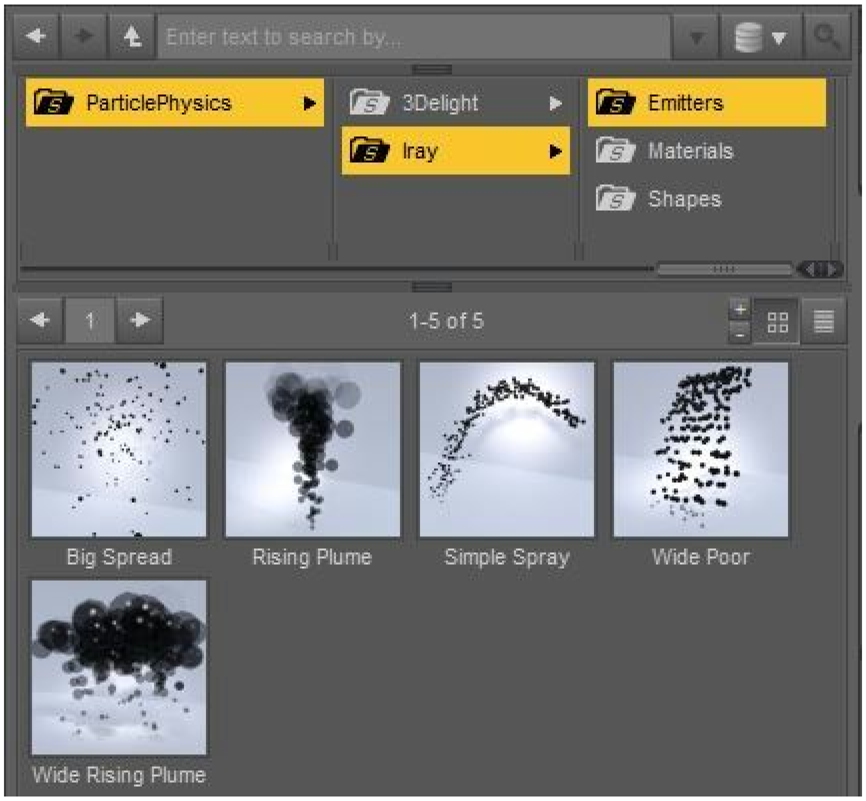
SimTenero Particle Physics

Getting Started

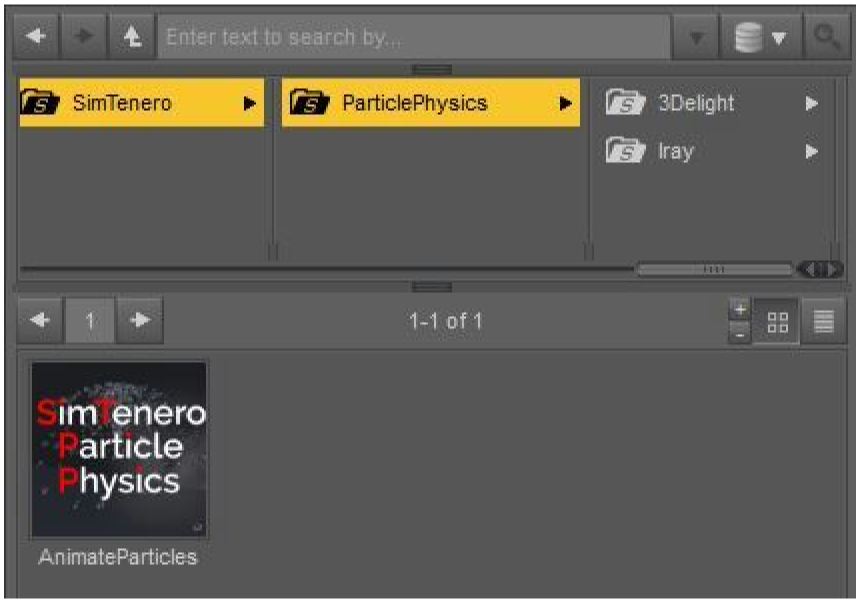
The heart of the particle system is the “Emitter.” This represents the point in space where particles will be created and contains all of the parameters that define a particle’s behavior.

Let’s start by loading an **Emitter Preset** into the Scene.

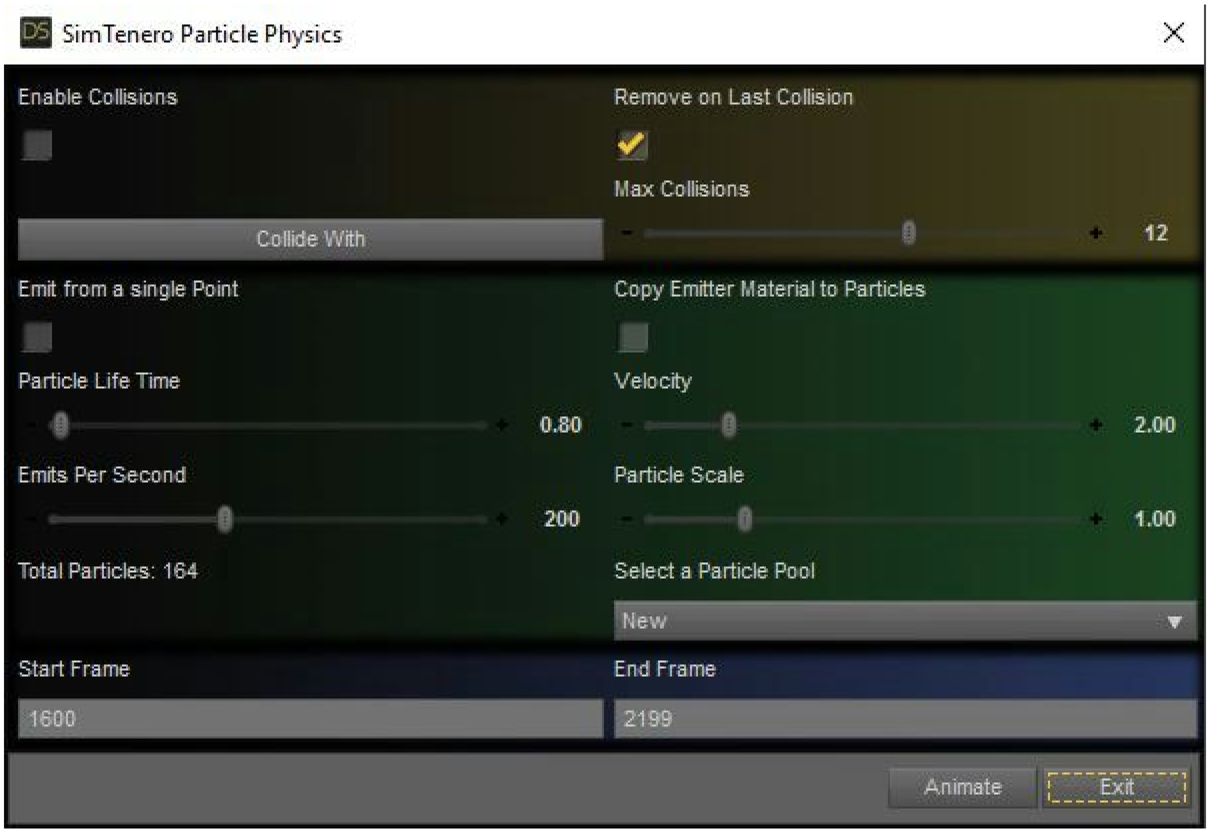


You should now see a red arrow in your scene. It’s size and shape will vary depending on the emitter you’ve chosen. Most, though not all, emitters are aligned to their “Local Space.” What that means is that particles will leave the emitter in whatever direction the arrow is pointing. Go ahead and point the arrow where you’ve like. You can even animate the arrow or attach it to another object in your scene!

That’s it, you are ready to animate! There’s a lot you can do with these emitters, which we’ll cover later on. For now, locate and run the “**AnimateParticles**” script in your content library.



Be sure you have a particle emitter selected before running the script. You'll be presented with several options related to your emitter, including collision options, but let's start with simple non-colliding particles first.



SimTenero Particle Physics

Enable Collisions ☐

Remove on Last Collision ✓

Max Collisions 12

Collide With

Emit from a single Point ☐

Copy Emitter Material to Particles ☐

Particle Life Time 0.80

Velocity 2.00

Emits Per Second 200

Particle Scale 1.00

Total Particles: 164

Select a Particle Pool: New

Start Frame: 1600

End Frame: 2199

Particle Life Time is the length of time, in seconds, that a particle will exist in the animation. When it's time is up, it will return to the particle pool. Longer times require more particles in your scene (you can see a real time update of "Total Particles" below this option). The more particles in your scene, the more time is needed to run the simulation, so be sure to set the Particle Life Time for only as long as you need. Try starting low, around 1 second, and doing a test simulation. If the particles aren't moving as far as you'd like, you can increase this number until you get the desired effect.

Emits Per Second is the number of particles that will be created during each second of the simulation. Just like Particle Life Time, this affects the number of total particles in your scene. As before, let's start with a modest number first, around 50 Emits Per Second.

The **Particle Pool** is a critical component of the simulation. When you run the simulation, the number of particles shown in the "Total Particles" section of the interface will be created in your scene and stored under a master node. This master node is a "Particle Pool." If you have existing pools in your scene, you can click the **Select a Particle Pool** dropdown and run a simulation on an existing pool. This will replace the particles animations from any previous simulations. This is very useful when you are testing and want to rerun the simulation as it saves you having to wait for a new pool to be created. For now, let's leave it on "New."

Particles will start emitting on the **Start Frame** and will finish emitting one Particle LifeTime before the **End Frame**. The range needs to be at least long enough to simulate one particle lifetime. What that means is, if your animation timeline is set to 30 Frames Per Second, and your Particle Lifetime is set to 2 seconds, you'll need need at least 60 frames in your range. It also helps to allow some time for the last few particles to finish up, so you may want to add some additional frames. Assuming you've set your Particle Lifetime to 1 second, let's set our Start Frame to 0 and our End Frame to 90. This should give the simulation plenty of time to do it's thing.

Now click **Animate** and watch the simulation unfold!

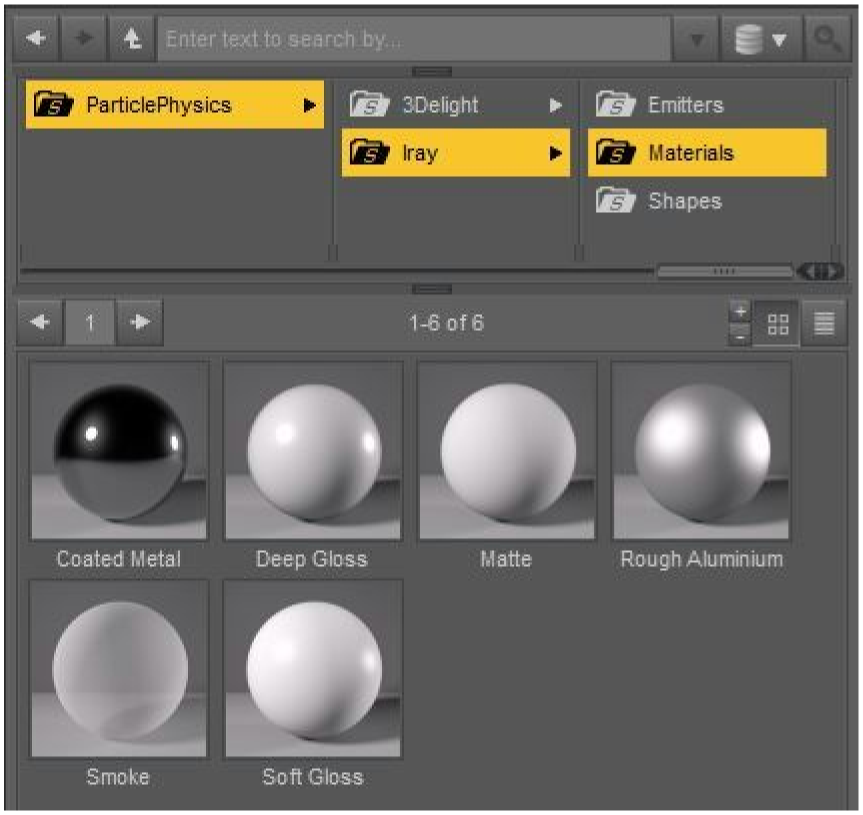
Materials and Shapes

Particle **Materials** function like any other material in Daz Studio, but there are a couple of important things to keep in mind. Both the material's Diffuse Color and Opacity will be overridden by the simulation. Modifying these two properties directly in the material can have undesirable results. To modify these parameters, select the Emitter in your scene and in the Parameters Pane modify *>SimTenero >Particles >Color >matColorStart* and *>SimTenero >Particles >Opacity >matOpacityStart*.

Particle **Shapes** represent the mesh used to represent each particle. Every particle is a single point in 3D space, but each is represented by a mesh, which allows them to be rendered in the scene. These shapes require special construction, so a valid SimTenero Particle Physics shape is required (2 are included with the core engine, while additional shapes are available as part of separate, Emitter Pack purchases).

It is important to note that both Materials and Shapes are applied to the Emitter itself prior to the simulation (though it is possible to select individual particles and modify their materials post-simulation).

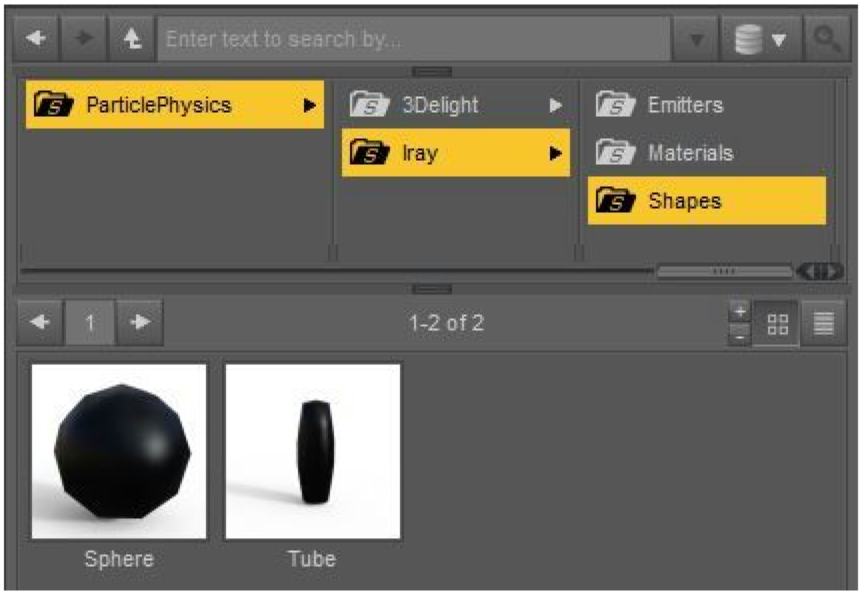
To apply a **Material Preset**, select an Emitter in your scene and load the preset from your content library.



You are not limited to the materials included with SimTenero Particle Physics. Many other material libraries will work, though it is not possible to guarantee compatibility, so some experimentation may be required. After applying a preset, you can select the Default Material of the Emitter to make further adjustments (bearing in mind the notes on Diffuse Color and Opacity mentioned above).

*If you have made material modifications, be sure to select **Copy Emitter Material to Particles** in the AnimateParticles interface.*

To apply a Shape Preset, select an Emitter in your scene and load a preset from your content library (only SimTenero Particle Physics Presets have the required construction to work with the simulation).

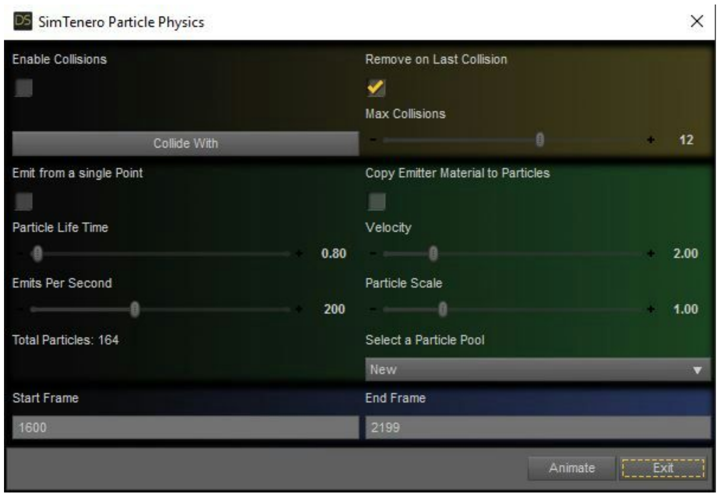


Shapes and Emitters from different packs can be combined, however this requires that those packs are stored installed to the same content library. For the best interoperability, it is strongly recommended to install the Core Engine and all Emitter Packs to the same content directory/library.

Shape Presets must be applied prior to simulation.

New Shapes cannot be applied to existing Particle Pools. To see the new shape, create a New Particle Pool.

Collision Settings



SimTenero Particle Physics

Enable Collisions: ☐
Collide With
Remove on Last Collision: ✓
Max Collisions: 12
Emit from a single Point: ☐
Copy Emitter Material to Particles: ☐
Particle Life Time: 0.80
Velocity: 2.00
Emits Per Second: 200
Particle Scale: 1.00
Total Particles: 164
Select a Particle Pool: New
Start Frame: 1600
End Frame: 2199
Animate
Exit

To enable your particles to collide with objects in your scene, select **Enable Collisions** in the AnimateParticles interface.

Next, click the **Collide With** button and place a checkmark next to each object you would like particles to collide with.

Because Daz Studio assets were not designed with collision in mind, please be aware that some assets will work better than others with the particle physics engine. See Optimizing Collisions below for more information.

Each particle will collide up to the number of times specified in the **Max Collisions** parameter.

If **Remove on Last Collision** is checked, particles will be returned to the pool when they reach the Max Collision limit, otherwise, they will continue to animate, but will no longer collide.

Bounce Loss is an *Advanced Parameter* defined in the Emitters *Parameters Pane >SimTenero >Particles >Collision >bounceLoss*. This figure represents the gravitational velocity that will be preserved on each collision. High values result in higher bounces, while lower values will result in progressively smaller bounces on each collision.

Optimizing Collisions

Collisions are the most processor-intensive element of the particle physics simulation. Performance is impacted by a number of variables, the most critical being the number of colliding objects in the scene, the number of colliding particles, and the polygon count of each colliding object.

SimTenero Particle Physics applies optimization routines to speed up collision detection, however, you will find in experimentation that some assets respond much more quickly than others. Below are two techniques that can greatly improve collision performance:

1. **Collision Primitives**
   Because collision performance is affected by the number of polygons in a colliding object, it can sometimes be useful to substitute primitives for complex asset meshes. To do this, simply create a new primitive shape that most closely resembles your target object by clicking on >Create >New Primitive in Daz Studio's main menu bar. Use the smallest number of sections possible while still maintaining the desired shape. Move and scale the new primitive so that it overlaps the target object. In the AnimateParticles interface, click Collide With, be sure to place a check by the new primitive, and be sure there is not a check beside the target object. Some examples:

   - Try using a cube in place of a coffee table or end table in your scene.
   - Try using a sphere in place of a figure's head.
   - Try using planes in place of room or environment walls.

2. **Daz Decimator**
   Daz Decimator is an add-on product available on the Daz Store. This technique is only available to users with Daz Decimator installed.
   Select the target object in your scene. In the Decimator tab, click Prepare to Decimate. Reduce the polygon count as far as possible while still maintaining the desired shape. Create a new LOD and verify the new LOD is applied in the object's Parameters Pane under Resolution. Now, simply select the object in the AnimateParticles interface as you normally would.

***A note on enclosed environments**: Some environment assets have walls constructed of individual object meshed, e.g. "Left Wall," Right Wall," "Floor," etc. These will typically perform better than environments that have walls, ceilings, and floors in a single mesh. In those instances, use of collision primitives (see above) rather than the original mesh geometry is strongly advised.*

Advanced Parameters

All of the parameters required by the simulation are stored in the Emitter's Parameters Pane. Modifications to these parameters can have unwanted or unexpected results and, so, their use is advised for advanced users who want further control over the simulation.

The Advanced Parameters may be familiar to users who have experience with physics simulations. Following is a brief overview of each section:

- **Collision** - These parameters are described in detail above. Most can be modified in either the Parameters Pane or in the AnimateParticles interface, with the exceoption of bounceLoss, which must be edited in the Parameters Pane, and Collide With, which must be edited in the AnimateParticles interface.
- **Color** - When colorChange is enabled, a particle will gradually change color from matColorStart to matColorFinish over its lifetime. colorStartChange determines where in the particle's life the change will begin.
- **Emission**
  - emitSpread is aligned to World Space, such that y always equals up, regardless of the emitters orientation.
  - Conversely, emitVelocity is aligned to the Emitter's Local Space. It is advisable that the total of the emitVelocity's XYZ values equals 1.
  - gravPerSecond describes the pull of gravity, negative values can be used for "anti-gravity" effects.
  - When randomAngle is enabled, each particle will emit with a random orientation, the degree of which is determined by the randomAngleMagnitude value.
  - When alignToDirection is enabled, each will be oriented along is current velocity vector. This setting overrides/disables randomAngle.

    - When singlePointEmit is enabled, all particles will emit from a single point in space. When it is disabled, particles will emit from a random location within the World Space bounds of the emitter.
    - velocityDampen represents the amount of a particle's velocity which is maintained over each frame of the simulation. Lower values will slow the particle over time. This setting is framerate dependant and may need to be adjusted if the target framerate is changed.

- **Main** - Contains the name of the target particle shape. This should only be modified using Emitter Presets.
- **Opacity** - Same function as Color, but alters the particle material's opacity over its lifetime.
- **Scale** - Same function as Color and Opacity, but alters the particle node's scale over its lifetime.