

- [skip to content](#)



## User Tools

- [Log In](#)

## Site Tools

Search

Tools ▼ >

Trace: • [ue4](#)

---

## Table of Contents

- [Building SpeedTree into the Engine](#)
- [The SpeedTree Applications](#)
- [Importing SpeedTrees into the Editor](#)
- [Using SpeedTree in a Level](#)
- [Additional Notes](#)



# Unreal Engine 4 Integration Details

SpeedTree has been integrated with Unreal Engine 4, and the integration code is available through Perforce from Epic.

The integration of SpeedTree with Unreal Engine 4 is a little different than the previous integration with UE3. In UE4, SpeedTree data is imported into a regular static mesh object. The SpeedTree SDK is only used in the UE4 Editor, while special UE4-specific runtime computation is built into the engine. This allows SpeedTree to be used in UE4 on any platform UE4 supports as easily as any other asset.

If you have any questions or comments, please don't hesitate to contact us at [support@speedtree.com](mailto:support@speedtree.com). We'll try to help you as best we can.

## Building SpeedTree into the Engine

SpeedTree is a module in Unreal Engine 4. It can be activated simply by placing the SpeedTree SDK in the correct spot and rebuilding the engine. The steps to get this working are as follows:

- Download the SpeedTree SDK
- If using the full version of the SpeedTree SDK, which includes no pre-built libraries, open the Core solution in the Source directory and batch build all configurations (Win32 and Win64).
- Rename the SpeedTree SDK directory "SpeedTree-v6.3"
- Place the renamed SpeedTree SDK in  
    <UE4>/Engine/Source/ThirdParty/NoRedist/SpeedTree/<SDK>
- Rebuild UE4

## The SpeedTree Applications

There is a version of the SpeedTree Applications specifically for creating trees to use in UE4. Basically, it streamlines the process so you don't have to worry about what UE4 can support when creating/tuning the tree. The UE4 version can load trees made elsewhere. If you do not have access to the UE4 version of the SpeedTree Applications, send us an [email](#), and we can make sure you get them.

To convert the artist's tree files (SPM files) to the runtime format that can be imported by the UE4 editor, you will use the SpeedTree Compiler. During compilation, the texture atlases will be created, billboards for the trees will be created and atlased, and optimized geometry for the trees will be generated and saved to SRT files. See the SpeedTree Compiler documentation for more information on this process.

You can use the standard SpeedTree for Games Applications to make trees for UE4 as well, but you will need to make sure to apply the same color set to all materials in the Modeler, set the wind mode to any of the presets besides "Custom", output TGA textures from the Compiler, and set the material merging mode in the Compiler to "SpeedTree 5.x mode". It is usually easier to use the UE4-specific applications.

# Importing SpeedTrees into the Editor



Once a tree has been compiled in the SpeedTree Compiler, you can import the resulting SRT file into the UE4 Editor. The SpeedTree model will be imported into a normal static mesh object.

You have a number of options during import that are presented to you in the import dialog. First is the choice between 3D geometry, billboards, or both. If a tree will only be used close-up, it is a “hero” tree, or you otherwise have trouble with the billboard as the lowest LOD (as is sometimes the case with dynamic shadows falling on the flat billboard geometry), you can choose to ignore the billboard mesh. Or, if a tree will only be used far away and you want to save loading the 3D geometry, you can only import the billboard. Just importing the billboard mesh also creates a static mesh that can be used with UE4's foliage layers.

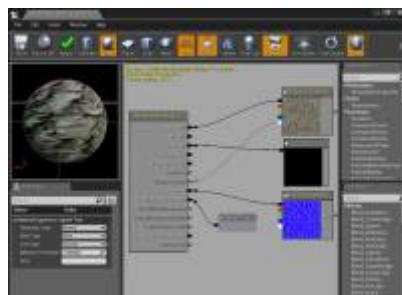
In the import dialog, you can also choose to include detail, specular, and branch seam blending in the materials used on the 3D tree. You can also add vertex processing to the materials, which puts a special SpeedTree node into the material that can handle wind, billboarding, and other effects computed per-vertex on the tree.

Finally, you can import any collision primitives that were set up in the SpeedTree Modeler into collision information on the new SpeedTree mesh.

## Using SpeedTree in a Level

Once the SpeedTree is imported, it is just like any other static mesh in how you handle it in the editor. You just drag it into the scene and place it however you want. SpeedTree meshes can be rotated, and they will maintain correct billboarding and wind effects.

To get wind working in a level, you will need to add a directional wind actor to the scene. Once you do this, SpeedTrees will react to the direction and strength set on that wind actor.



At any time, you can modify the materials created by the importer. The SpeedTree node hooked to

the world position offset is what handles wind effects, billboarding, and smooth LOD.

If you wish to disable wind effects to optimize the rendering, you can do so here. Note that the wind style (besides “None”) must match how the tree was tuned in the Modeler. If you delete the SpeedTree node entirely, the SpeedTree will basically become a regular static mesh.

## Additional Notes

At this time, smooth LOD on SpeedTrees is disabled due to some complexity in the rendering of static meshes. LOD will pop similarly to other static meshes. There are plans to get this working in the future.

---

[Read our blog >>](#)

- [Home](#)
- [Company](#)
- [3D Animation Software](#)
- [3D Tree/Plant Library](#)
- [Accolades](#)
- [Documentation](#)
- [Contact](#)
- [Privacy Policy](#)
- [Terms & Conditions](#)
- [Site Map](#)
- ©2017 IDV, Inc. All Rights Reserved.
- [Questions?](#)

