Navigation Mesh

Reference for the Navigation Mesh section of the Unreal Engine Project Settings.

Navigation Mesh

Display

Section	Description
Draw Triangle Edges	Draw edges of every NavMesh's triangle.
Draw Poly Edges	Draw edges of every poly (that is, not only border-edges).
Draw Filled Polys	If disabled, skips filling drawn NavMesh polygons.
Draw NavMesh Edges	Draw border-edges.
Draw Tile Bounds	Draw the tile boundaries.
Draw Path Colliding Geometry	Draw input geometry, passed to the NavMesh generator. We recommend disabling other geometry rendering via viewport showflags in editor.
Draw Tile Labels	Draw the labels of the tiles.
Draw Polygon Indices	Draw a label for every poly that indicates its poly and tile indices.
Draw Polygon Costs	Draw a label for every poly that indicates its default and fixed costs.
Draw Polygon Flags	Draw a label for every poly that indicates its poly and area flags.
Draw Labels on Path Nodes	Draw labels on path nodes.

Section	Description
Draw Nav Links	Draw navigation links.
Draw Failed Nav Links	Draw failed navigation links.
Draw Clusters	Draw NavMesh's clusters and cluster links (requires `WITH_NAVMESH_CLUSTER_LINKS=1`).
Draw Octree	Draw edges of every NavMesh's triangle.
Draw Octree Details	Draw octree details.
Draw Marked Forbidden Polys	Draw marked forbidden polys.
Draw Offset	Vertical offset added to NavMesh's debug representation for better readability.

Tile Generation Debug

Section	Description
Tile Generation Debug Enabled	If set, the selected internal debug data will be kept during tile generation to be displayed with the NavMesh.
Tile Coordinate	Selected tile will have its internal data kept. Displaying the NavMesh using **Draw Tile Labels** shows tile coordinates.
Heightfield Solid from Rasterization	Heightfield solid from rasterization.
Heightfield Solid Post Radius Filtering	Heightfield solid post radius filtering.
Heightfield Solid Post Height Filtering	Height solid post height filtering.
Compact Heightfield	Compact heightfield.
Compact Heightfield Eroded	Compact heightfield eroded.
Compact Heightfield Regions	Compact heightfield regions.

Section	Description
Compact Heightfield Distances	Compact heightfield distances.
Tile Cache Layer Areas	Tile cache layer areas.
Tile Cache Layer Regions	Tile cache layer regions.
Tile Cache Contours	Tile cache contours.
Tile Cache Poly Mesh	Tile cache poly mesh.
Tile Cache Detail Mesh	Tile cache detail mesh.

Section	Description
Enable Drawing	If enabled, built Navigation Data will display in the Level when the Show Navigation flag is enabled in Editor Preferences. By default, this can be toggled with by pressing the **P** key.

Generation

Section	Description
Fixed Tile Pool Size	If enabled, the NavMesh will allocate a fixed size pool for tiles. Should be enabled to support streaming.
Tile Pool Size	Maximum number of tiles NavMesh can hold.
Tile Size UU	The size of a single navigation tile, expressed in Unreal Units (1 UU = 1 cm).
Cell Size	Horizontal size of a voxelization cell.
Cell Height	Vertical size of a voxelization cell.
Agent Radius	Radius of the smallest agent to traverse this NavMesh.
Agent Height	Height of the tallest agent that will be able to find paths using this NavMesh.
Agent Max Slope	The maximum slope (angle) that the agent can move

Section	Description
	on.
Agent Max Step Height	Largest vertical step the agent can perform.
Min Region Area	The minimum dimension of an area. Areas smaller than this will be discarded.
Merge Region Size	The size limit of regions to be merged with bigger regions (watershed partitioning only).
Max Simplification Error	Defines how much navigable shapes can get simplified. Higher values mean more freedom and fewer constraints.
Sort Navigation Areas by Cost	Controls whether Navigation Areas will be sorted by cost before application to NavMesh during NavMesh generation. This is relevant when there are areas overlapping and you want to have area cost express area relevancy as well. Enabling this option will result in having the area sorted by cost, but it will also increase NavMesh generation cost by a small amount.
Is World Partitioned	In a world-partitioned map, defines whether this NavMesh is using world partitioning.
Max Simultaneous Tile Generation Jobs Count	Sets the maximum limit for the number of asynchronous tile generators running at one time. Also used for some synchronous tasks.
Tile Number Hard Limit	Absolute hard limit to the number of NavMesh tiles. Be very, very careful when modifying this setting while having big maps with NavMesh. A single, empty tile takes 176 bytes and empty tiles are allocated up front (subject to change in future releases). `TileNumberHardLimit` is always rounded up to the closest power of 2.

Section	Description
Poly Ref Tile Bits	Poly ref tile bits.
Poly Ref Nav Poly Bits	Poly ref nav poly bits.
Poly Ref Salt Bits	Poly ref salt bits.
NavMesh Origin Offset	Set this value if you don't want your tiles to start at `(0,0,0)`.
Region Partitioning	Partitioning method for creating NavMesh polys. You can choose from the following options: • **Monotone** • **Watershed** • **Chunky Monotone**
Layer Partitioning	Partitioning method for creating tile layers. You can choose from the following options: • **Monotone** • **Watershed** • **Chunky Monotone**
Region Chunk Splits	Determines how many chunks are used to divide the current region along each axis when you select the **Chunky Monotone** option in the **Region Partitioning** setting.
Layer Chunk Splits	Determines how many chunks are used to divide the current layer along each axis when you select the **Chunky Monotone** option in the **Layer Partitioning** setting.
Perform Voxel Filtering	Controls whether voxel filtering will be applied (via `FRecastTileGenerator::ApplyVoxelFilter`). Results in the generated NavMesh better fitting navigation bounds, but slightly lowers generation performance.
Mark Low Height Areas	Mark areas with insufficient free height above instead of cutting them out (accessible only for area modifiers using replace mode).
Use Extra Top Cell when Marking Areas	Expand the top of the area nav modifier's bounds by one cell height when applying to the NavMesh.

Section	Description
Filter Low Span Sequences	If set, only a single low height span will be allowed under a valid one.
Filter Low Span from Tile Cache	If set, only low height spans with corresponding area modifier will be stored in tile cache (reduces memory, can't modify without full tile rebuild).
Do Fully Async Nav Data Gathering	If set, NavMesh data gathering will never happen on the game thread and will only be done on background threads.

Query

Section	Description
Heuristic Scale	Euclidean distance heuristic scale used while pathfinding.
Vertical Deviation from Ground Compensation	Value added to each search height to compensate for the error between NavMesh polys and walkable geometry.

Runtime

Section	Description
Force Rebuild on Load	By default, navigation will skip the first update after being successfully loaded. Setting `bForceRebuildOnLoad` to false can override this behavior.
Auto-Destroy when No Navigation	Defines whether this instance should destroy itself if there's no navigation system in the world when the world gets created or loaded.
Runtime Generation	Navigation Data runtime generation options. You can choose from the following options: • **Static** • **Dynamic Modifiers Only** • **Dynamic**

Section	Description
Observed Paths Tick Interval	All observed paths will be processed every `ObservedPathsTickInterval` second.
Can Be Main Nav Data	If set, Navigation Data can act as default one in navigation system's queries.
Can Spawn on Rebuild	If set, Navigation Data will be spawned in the persistent Level during rebuild if the Actor doesn't exist.

Tick

Section	Description
Allow Tick Before Begin Play	Defines whether you allow this Actor to tick before it receives the `BeginPlay` event. Normally, Actors shouldn't tick until after `BeginPlay`. This setting allows this behavior to be overridden. This Actor must be able to tick for this setting to be relevant.

Collision

Section	Description
Generate Overlap Events During Level Streaming	If true, this Actor will generate overlap Begin / End events when spawned as part of level streaming, which includes initial level load. You might enable this if a streaming level loads around an Actor and you want Begin / End overlap events to trigger. See the **Update Overlaps Method During Level Streaming** setting in this section.
Update Overlaps Method During Level Streaming	Condition for calling `UpdateOverlaps()` to initialize overlap state when loaded in during level streaming. If set to `UseConfigDefault`, the default specified in `.ini` (displayed in `DefaultUpdateOverlapsMethodDuringLevelStreaming`) will be used. If overlaps are not initialized, this Actor and attached Components will not have an initial state of what objects are touching it, and overlap events may only come in once one of those objects update overlaps themselves (for example, when moving).

Section	Description
	However, if an object touching it does initialize the state, both objects will know about their touching state with each other. This can be a potentially large performance-saving improvement during level loading and streaming, and is safe if the object and others initially overlapping it do not need the overlap state because they will not trigger overlap notifications. You can choose from the following options: - **Use Config Default** - **Always Update** - **Only Update Movable** - **Never Update** ⓘ If `bGenerateOverlapEventsDuringLevelStreaming` is true, overlaps are always updated in this case, but that flag determines whether the Begin / End overlap events are triggered. See `bGenerateOverlapEventsDuringLevelStreaming`, `DefaultUpdateOverlapsMethodDuringLevelStreaming`, `GetUpdateOverlapsMethodDuringLevelStreaming()`.

Default Update Overlaps Method During Level Streaming	Default value taken from the config file for this class when `UseConfigDefault` is chosen for `UpdateOverlapsMethodDuringLevelStreaming`.
	This allows a default to be chosen per class in the matching config.
	For example, for Actor, it could be specified in `DefaultEngine.ini` as:
	`[/Script/Engine.Actor]`
	`DefaultUpdateOverlapsMethodDuringLevelStreaming =`
	Copy full snippet
	Another subclass could set their default to something different, such as:
	`[/Script/Engine.BlockingVolume]`
	`DefaultUpdateOverlapsMethodDuringLevelStreaming =`
	Copy full snippet

Section	Description
	See `UpdateOverlapsMethodDuringLevelStreaming`.
Relevant for Level Bounds	If true, this actor's component's bounds will be included in the level's bounding box unless the Actor's class has overridden `IsLevelBoundsRelevant`.

HLOD

Section	Description
Include Actor in HLOD	Specifies whether this Actor should be considered during HLOD generation.
HLOD Layer	The UHLODLayer in which this Actor should be included.

World Partition

Section	Description
Runtime Grid	Determines in which partition grid this actor will be placed in the partition (if the world is partitioned). If set to `None`, the decision will be left to the partition.
Is Spatially Loaded	Determines whether this actor is spatially loaded when placed in a partitioned world. If true, this actor will be loaded when in the range of any streaming sources and if (1) in no data layers, or (2) one or more of its data layers are enabled. If false, this actor will be loaded if (1) in no data layers, or (2) one or more of its data layers are enabled.

Cooking

Section	Description
Is Editor Only Actor	Defines whether this actor is editor only. Use with care, as if this actor is referenced by anything else, that reference will be `NULL` in cooked builds.

Section	Description
Generate Optimized Blueprint Component Data	Defines whether to cook additional data to speed up spawn events at runtime for any Blueprint classes based on this Actor. This option may slightly increase memory usage in a cooked build.

Data Layers

Section	Description
Data Layers	`DataLayers` the actor belongs to.