EQS Node Reference: Tests

Describes how to use Tests within EQS to generate the "best" option.

Learn to use this **Experimental** feature, but use caution when shipping with it.

Within the Environmental Query System (EQS), a **Test** can be performed to determine which **Item** produced from a Generator is the "best" option, given the Context (or frame of reference). Several Tests are provided with the Engine that covers a good percentage of use cases, such as "can an Item trace (see) another Location" or "is the distance between and Item and its Context within a specified range". You can add multiple Tests to a Generator which can be an effective way to narrow down the results, giving you the best possible option.

If the default Engine Tests do not accomplish what you desire, you can create custom Tests through C++ code.

Common Test Properties

Each Test type has some unique properties to it that enables you to define how the Test is performed. However, for all Tests, some common properties are used to define what the purpose of the Test is and what to do with the results. For example, is the Test used for filtering out results, or is it being used to score results and weight them, or a combination of both? You can define the **Test Purpose** among other properties from the **Details** panel when selecting the Test in the EQS Editor.

Test Properties

Property	Description
Test Comment	Optional comment or explanation about what the Test is used for. This is useful when the purpose of the Test may not be clear, especially when there are multiple Tests of the same type.
Test Purpose	This defines what additional options are available in the Test and what the test should be used for. • **Filter Only**: Used to filter possible results. Items that fail the Test will be removed. • **Score Only**: Used to score possible results. Items returned are given a weight value. • **Filter and Score**: Used to filter and score results.

Filter Properties

The following options are available when **Test Purpose** is set to **Filter** (or set to **Filter and Score**):

> Filtering is done before scoring to avoid calculating the score on filtered-out items.

Property	Description
Bool Match	This is the value (true or false) to match to grant a scoring factor. When performing a Test, not matching this value will not

Property	Description
	change the score. For example, on a **Trace** Test, true or false did we hit something? Or for **Pathing** does a path exist?
Multiple Context Filter Op	Defines the filtering operator when the **Distance To** Context returns multiple items. **All Pass** means all Contexts must pass while **Any Pass** states at least one Context must pass.
Float Value Min	Filter any value less than or equal to this value. ⓘ This option is only available for **Distance** and **Dot** Tests.
Float Value Max	Filter any value greater than or equal to this value. ⓘ This option is only available for **Distance** and **Dot** Tests.
Filter Type	Used to change the type of Filter applied to **Minimum**, **Maximum**, or a **Range** of values. Any values outside the values specified in the **Float Value Min** and/or **Float Value Max** properties will be culled out. ⓘ This option is only available for **Distance** and **Dot** Tests.

Score Properties

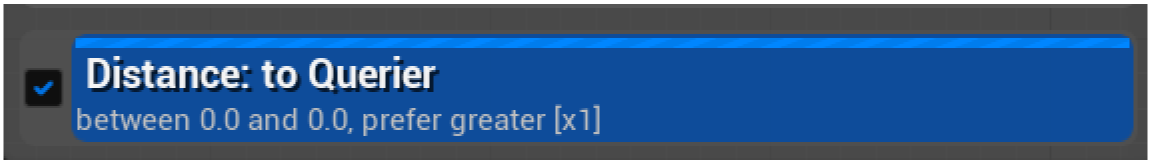
The following options are available when **Test Purpose** is set to **Score** (or set to **Filter and Score**).

Property	Description
Multiple Context Filter Op	Defines the filtering operator when the **Distance To** Context returns multiple items. **All Pass** means all Contexts must pass while **Any Pass** states at least one Context must pass.

Property	Description
****Clamp Min Type****	Defines whether a **Specified Value** should be used to normalize the raw test value before applying the scoring formula, or if the **lowest** value found (Tested) should be used. ⓘ This option is only available for **Distance** and **Dot** Tests.
Clamp Max Type	Defines whether a **Specified Value** should be used to normalize the raw test value before applying the scoring formula, or if the **highest** value found (Tested) should be used. ⓘ This option is only available for **Distance** and **Dot** Tests.
Scoring Equation	This modifies the score of the Test to adhere to a curve of the **Constant**, **Linear**, **Square**, **Inverse Linear**, or **Square Root** type. ⓘ This option is only available for **Distance** and **Dot** Tests.
Scoring Factor	The weight (factor) in which to multiply the normalized score after the scoring equation is applied. This value can be a negative number. ⓘ This option is only available for **Distance** and **Dot** Tests.
Normalization Type	Specifies how to determine the value span used to normalize score. **Absolute** (use 0 as the base of normalization range) or **Relative to Scores** (use the lowest Item score as the base of normalization range). ⓘ This option is only available for **Distance** and **Dot** Tests.

Property	Description
Reference Value	Used to normalize Test's results in such a way that the closer a value is to **Reference Value** the higher normalized result it will produce. The value farthest away from **Reference Value** will be normalized to 0, and all other values in between will get normalized linearly with the distance to **Reference Value**. ⓘ This option is only available for **Distance** and **Dot** Tests.

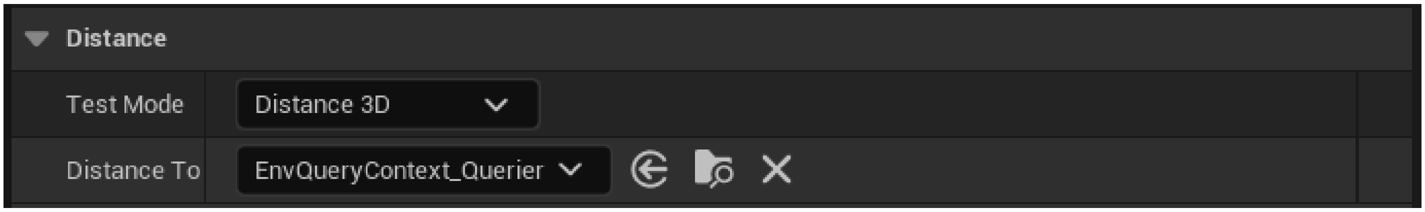
Distance



☑ Distance: to Querier

between 0.0 and 0.0, prefer greater [x1]

The **Distance** Test will return the direct distance between the Item and the chosen **Distance To** property. If Distance To is more than one location, it averages the results of each distance check.

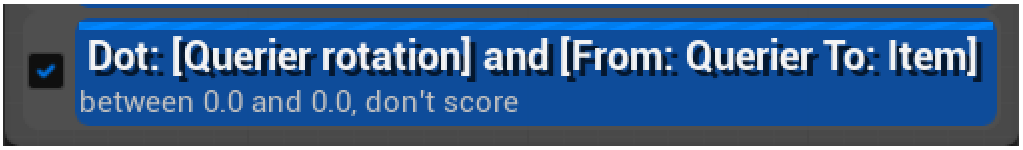


▼ Distance

Test Mode	Distance 3D
Distance To	EnvQueryContext_Querier

Property	Description
Test Mode	The method used to test Distance: in 3D space, in 2D as an XY plane, along the Z or Z (absolute) axis.
Distance To	The Context that will be used to measure the distance to.

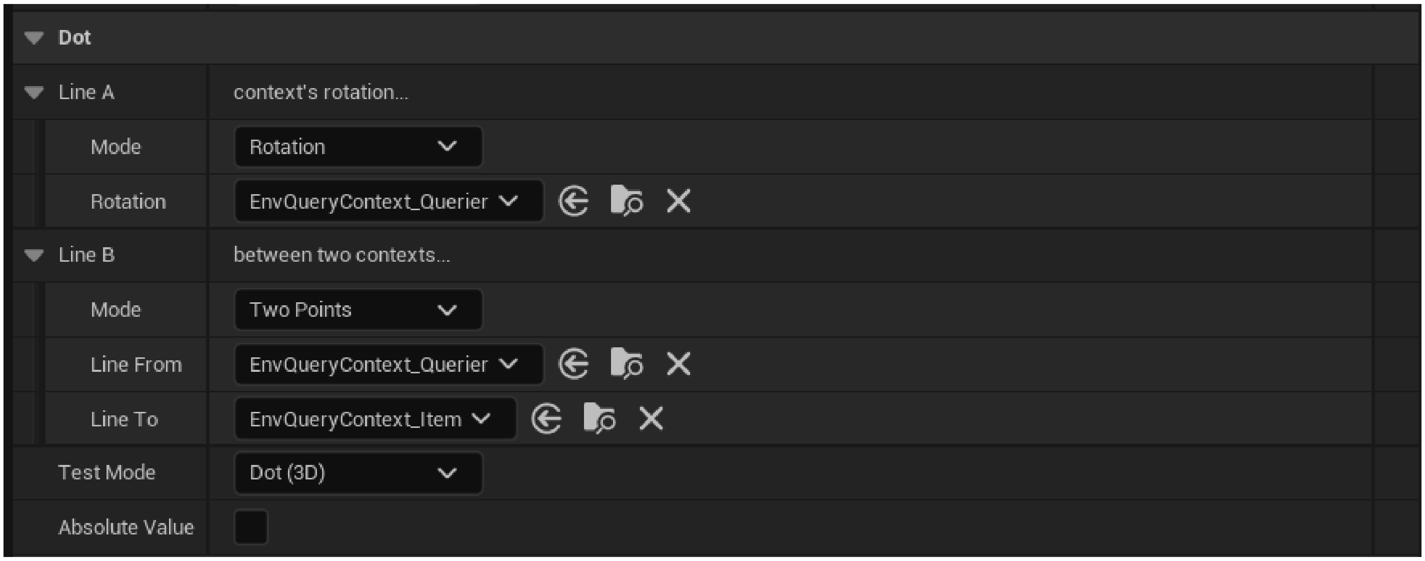
Dot



Dot: [Querier rotation] and [From: Querier To: Item]

between 0.0 and 0.0, don't score

The **Dot** Test calculates the Dot Product of two vectors. These can be Context rotations or vector from one point to another. This Test is useful for determining if something is facing something else.



Dot	
Line A	context's rotation...
Mode	Rotation
Rotation	EnvQueryContext_Querier
Line B	between two contexts...
Mode	Two Points
Line From	EnvQueryContext_Querier
Line To	EnvQueryContext_Item
Test Mode	Dot (3D)
Absolute Value	□

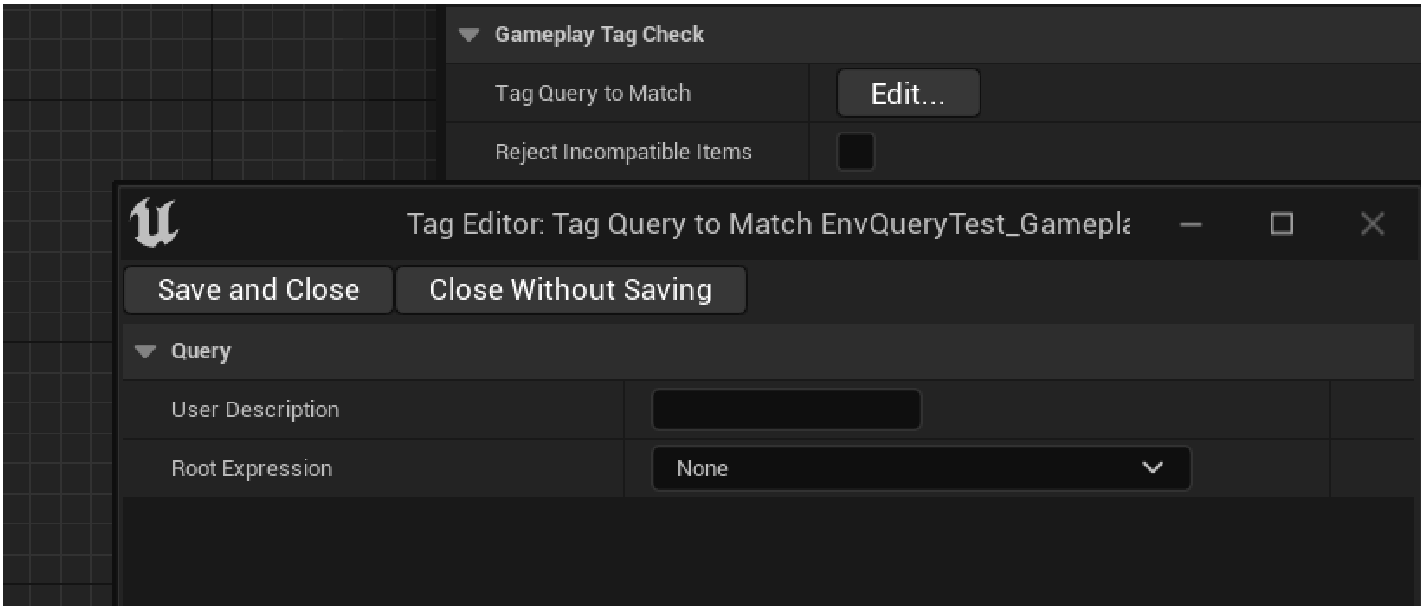
The following properties are available for the **Dot** Test:

Property	Description
Line A Mode	This is used to define the direction of the **first** line used by the Test. There are two methods you use to obtain the direction: • **Rotation**: The specified Context will be used as a direction. • **Two Points**: Direction from Location of one Context to another.
Line B Mode	This is used to define the direction of the **second** line used by the Test. There are two methods you can use to obtain the direction: • **Rotation**: The specified Context will be used as a direction. • **Two Points**: Direction from Location of one Context to another.
Test Mode	Whether the Test should calculate using the complete 3D vector or just the 2D heading vector of the **Line A** and **Line B** vectors.

Property	Description
Absolute Value	This will make the Test return the Absolute Value of the Dot Product (a Dot Product returns a value from -1.0 to 1.0).

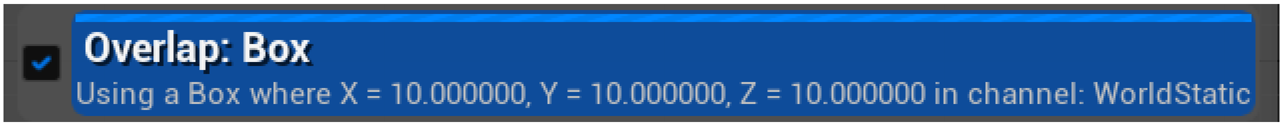
Gameplay Tags

The **Gameplay Tags** Test enables you to specify a Tag to query and attempt to match in your Test.



Property	Description
Tag Query to Match	Opens the Gameplay Tags Editor where you can specify the Tags to validate against.
Reject Incompatible Items	Controls how to treat actors that do not implement IGamePlayTgAssetInterface. If true, actors that do not implement the interface will be ignored, meaning they will not be scored and will not be considered when filtering. If false, actors that do not implement the interface will be included in filter and score operations with a zero score.

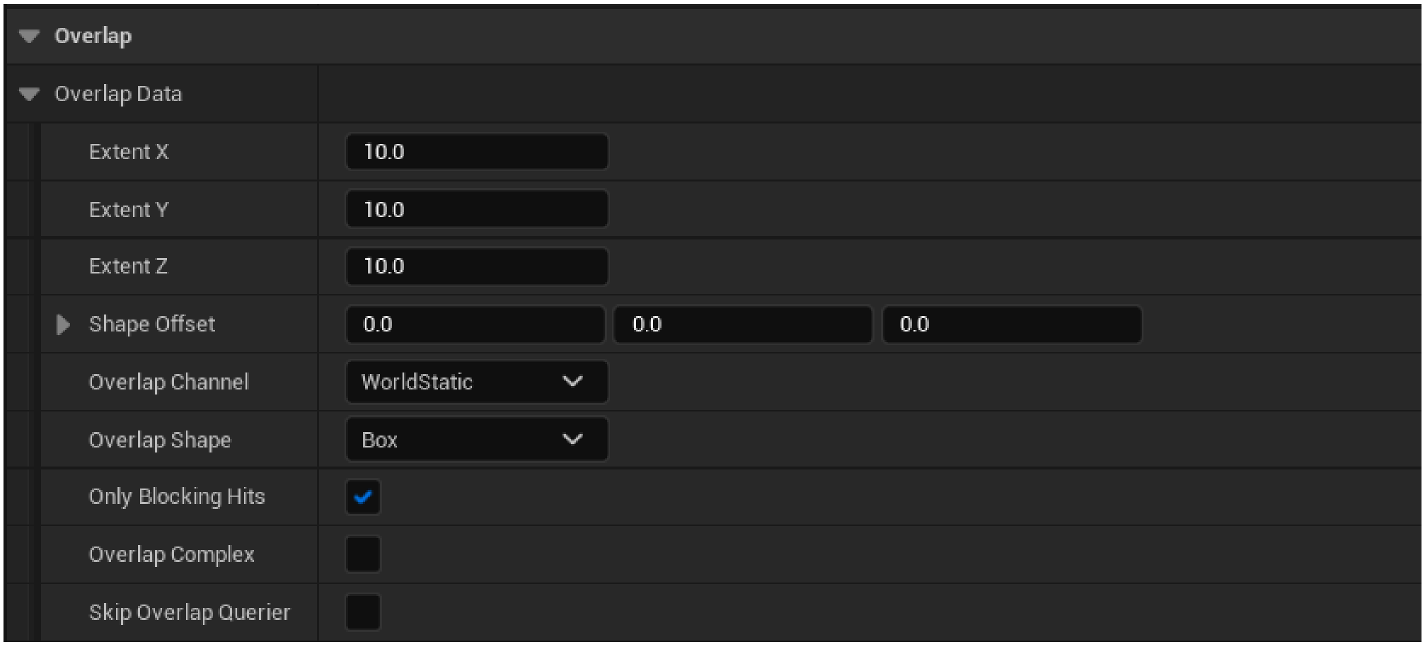
Overlap



☑ Overlap: Box

Using a Box where X = 10.000000, Y = 10.000000, Z = 10.000000 in channel: WorldStatic

The **Overlap** Test can be used to determine if an Item (or Items) are within the bounds defined in the properties.



Overlap

Overlap Data

Extent X: 10.0

Extent Y: 10.0

Extent Z: 10.0

Shape Offset: 0.0 0.0 0.0

Overlap Channel: WorldStatic

Overlap Shape: Box

Only Blocking Hits: ☑

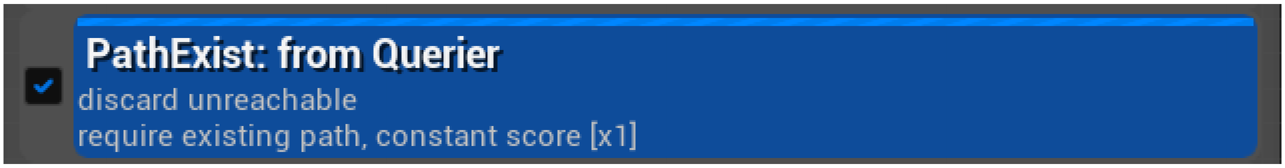
Overlap Complex: □

Skip Overlap Querier: □

Property	Description
Extent X	Shape parameter for the overlap along the X-axis.
Extent Y	Shape parameter for the overlap along the Y-axis.
Extent Z	Shape parameter for the overlap along the Z-axis.
Shape Offset	Offset from the Item location at which to test the overlap. For example, you may need to offset vertically to avoid overlaps with flat ground.
Overlap Channel	Geometry trace channel used for overlap.
Overlap Shape	The shape used for geometry overlap (**Box**, **Sphere**, or **Capsule**).

Property	Description
Only Blocking Hits	if set, overlap will only look for blocking hits.
Overlap Complex	If set, overlap will only run on complex collisions.
Skip Overlap Querier	If set, overlap will skip querier context hits.

Pathfinding

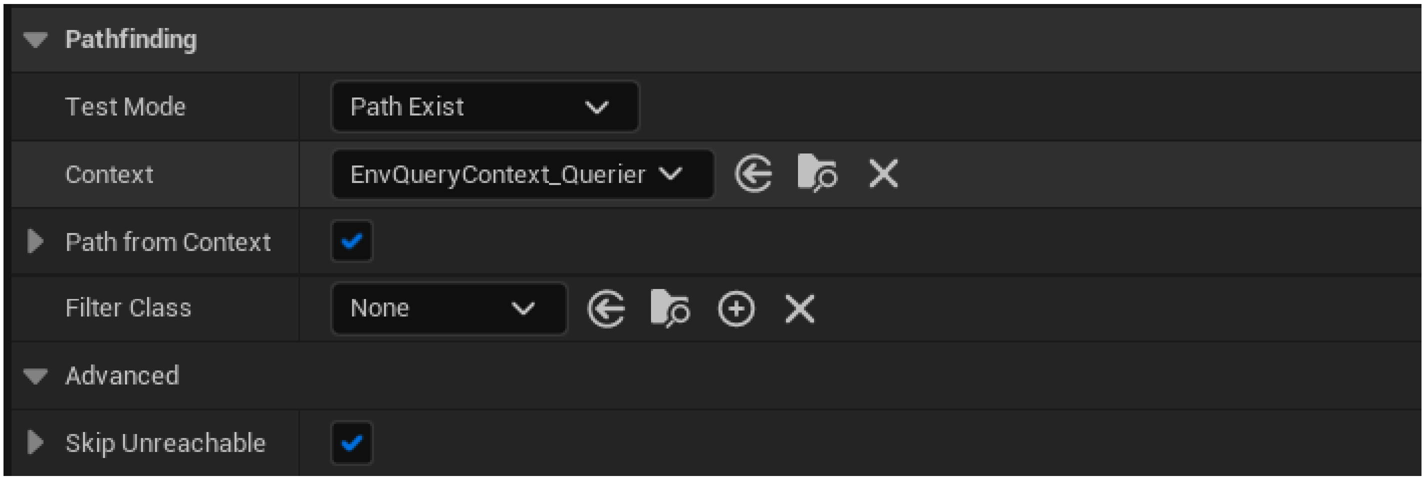


☑ **PathExist: from Querier**

discard unreachable

require existing path, constant score [x1]

The **Pathfinding** Test can be used to determine if a path exists to (or from) the Context, how expensive the path to (or from) the Context is, or how long the path is.



Pathfinding

Test Mode	Path Exist
Context	EnvQueryContext_Querier
Path from Context	☑
Filter Class	None

Advanced

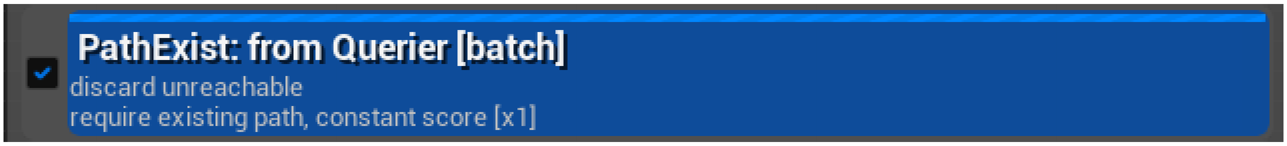
Skip Unreachable	☑

Property	Description
Test Mode	The method in which to apply the Test: • **Path Exist**: Does the path exist to (or from) the Context. • **Path Cost**: How expensive is the path to (or from) the Context. • **Path Length**: How long is the path to (or from) the Context.
Context	This is the Context of the path to or from.

Property	Description
Path from Context	Should the pathfinder go to (false) or from (true) the Context.
Filter Class	The optional navigation filter to use in pathfinding.
Skip Unreachable	If set, Items with failed paths will be invalidated.

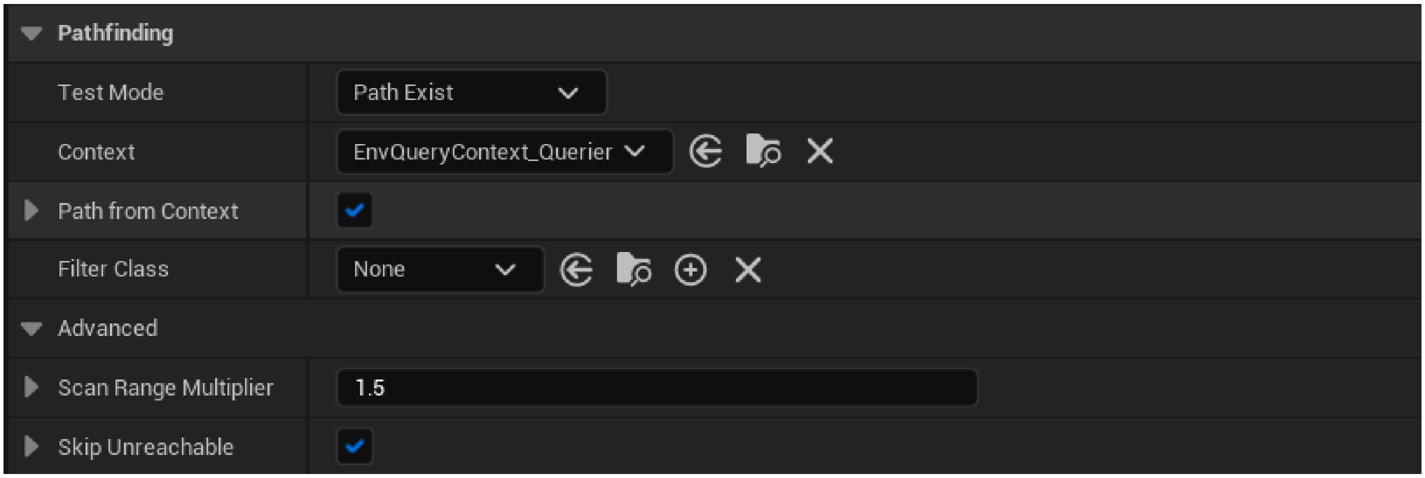
When **Test Mode** is set to **Path Cost** or **Path Length**, the **Filter** and **Score** sections of the **Details** panel change to provide the options typically only available to the common properties outlined for **Dot** and **Distance** tests.

Pathfinding Batch



☑ PathExist: from Querier [batch]
discard unreachable
require existing path, constant score [x1]

The **Pathfinding** Test can be used to determine if a path exists to (or from) the Context, how expensive the path to (or from) the Context is, or how long the path is. Every processed Context (paths) will be scored depending on the defined **Test Mode**.



Pathfinding

Test Mode: Path Exist

Context: EnvQueryContext_Querier

Path from Context: ☑

Filter Class: None

Advanced

Scan Range Multiplier: 1.5

Skip Unreachable: ☑

Property	Description
Test Mode	The method in which to apply the Test: - **Path Exist**: Does the path exist to (or from) the Context.

Property	Description
	• **Path Cost**: How expensive is the path to (or from) the Context. • **Path Length**: How long is the path to (or from) the Context.
Context	This is the Context that the AI should path to or from.
Path from Context	Should the pathfinder go to (false) or from (true) the Context.
Filter Class	The optional navigation filter to use in pathfinding.
Scan Range Multiplier	The multiplier for the max distance between point and Context.
Skip Unreachable	If set, Items with failed paths will be invalidated.

When **Test Mode** is set to **Path Cost** or **Path Length**, the **Filter** and **Score** sections of the **Details** panel change to provide the options typically only available to the common properties outlined for **Dot** and **Distance** tests.

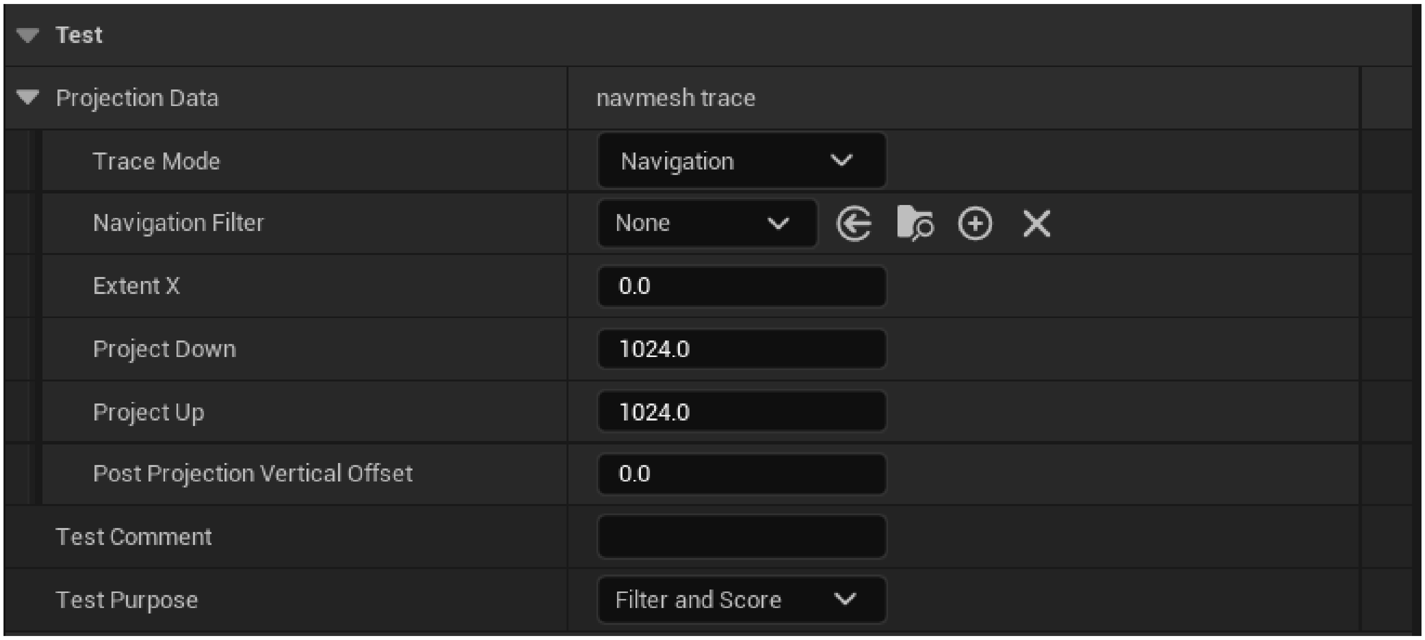
Project

☑ Project

require projected, constant score [x1]

The **Project** Test can be used to project the resulting Items onto the NavMesh (and which NavMesh data set to use).

This will move Items that may be inside walls or blocked, back onto the NavMesh, which can create bunching if a grid line happens to be just beyond the edge of the NavMesh.



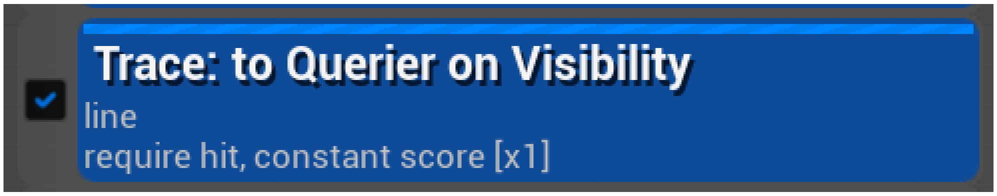
Test

Projection Data	navmesh trace
Trace Mode	Navigation
Navigation Filter	None
Extent X	0.0
Project Down	1024.0
Project Up	1024.0
Post Projection Vertical Offset	0.0
Test Comment	
Test Purpose	Filter and Score

Property	Description
Trace Mode	This is the shape used for geometry tracing: • **Navigation**: Does the path exist to (or from) the Context. • **Geometry by Channel**: Determines how expensive the path is to (or from) the Context by using channel tracing. • **Geometry by Profile**: Determines how expensive the path is to (or from) the Context by using profile tracing.
Navigation Filter	The (optional) navigation filter class to used.
Extent X	Shape parameter for trace.
Project Down	Search height is defined below the specified point.
Project Up	Search Height is defined above the specified point.
Post Projection Vertical Offset	This value will be added to the resulting location's Z-axis.

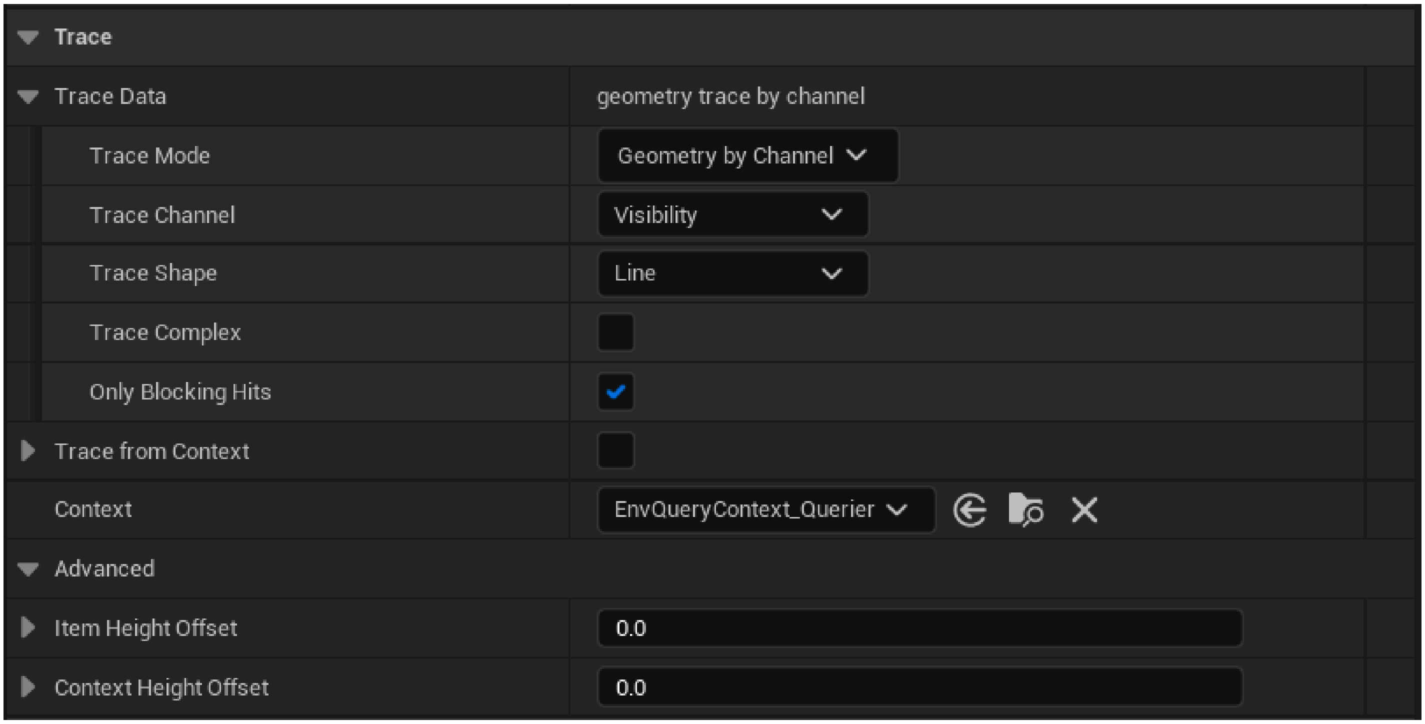
Property	Description
	ⓘ This can be useful when projecting points on navigation since the Navmesh is just an approximation of Level geometry and items may end up being under collide-able geometry which would, for example, falsify visibility tests.

Trace



☑ Trace: to Querier on Visibility
line
require hit, constant score [x1]

The **Trace** Test will trace to (or from) an Item or Context and return if it hit it or not something. You can invert the result using the **Filter** option, **Bool Match**. One typical use-case for this type of test is to determine if an enemy can (or cannot) see a Player in the Level.



Trace	
Trace Data	geometry trace by channel
Trace Mode	Geometry by Channel
Trace Channel	Visibility
Trace Shape	Line
Trace Complex	☐
Only Blocking Hits	☑
Trace from Context	☐
Context	EnvQueryContext_Querier
Advanced	
Item Height Offset	0.0
Context Height Offset	0.0

Property	Description
Trace Mode	This is the shape used for geometry tracing:

Property	Description
	- **Geometry by Channel**: Determines how expensive the path is to (or from) the Context by using channel tracing. - **Geometry by Profile**: Determines how expensive the path is to (or from) the Context by using profile tracing.
Trace Channel	This is the channel to perform the trace against. By default, **Visibility** and **Camera** are the available options however, additional channels can be added in the **Edit Menu > Project Settings > Physics > Trace Channels** section of the Project Settings.
Trace Shape	The shape to perform the trace: **Line**, **Sphere**, **Box**, or **Capsule**.
Trace Complex	Whether the trace should be against the mesh (complex) or just the simple collision.
Only Blocking Hits	Whether the trace uses blocking or non-blocking traces in its results.
Trace from Context	The Context to trace from, such as the Querier, an Item, or any custom Contexts you may have created.
Context	This is the other end of the trace.
Item Height Offset	This will add a Z offset, in cm, to the Item the Test is tracing to (or from).
Context Height Offset	This will add a Z offset, in cm, to the Context the test is tracing to (or from).