

- Developer
- / Documentation
- / Unreal Engine ▾
- / Unreal Engine 5.4 Documentation
- / Making Interactive Experiences
- / Artificial Intelligence
- / Behavior Trees
- / Behavior Tree Node Reference
- / Behavior Tree Node Reference: Tasks

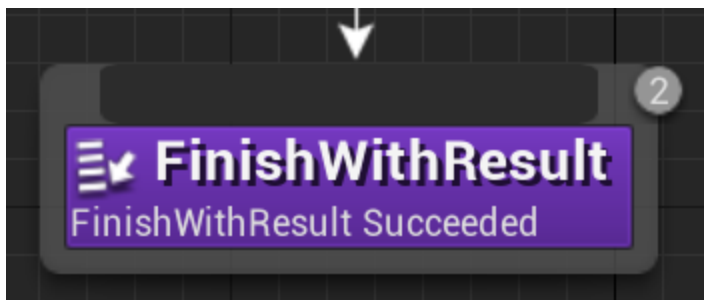
# Behavior Tree Node Reference: Tasks

Reference for the Behavior Tree Task nodes.



This is a reference page for the **Task** nodes available in the Behavior Tree Editor. Tasks are nodes that "do" things, like move an AI, or adjust Blackboard values. They can have [Decorators](#) or [Services](#) attached to them.

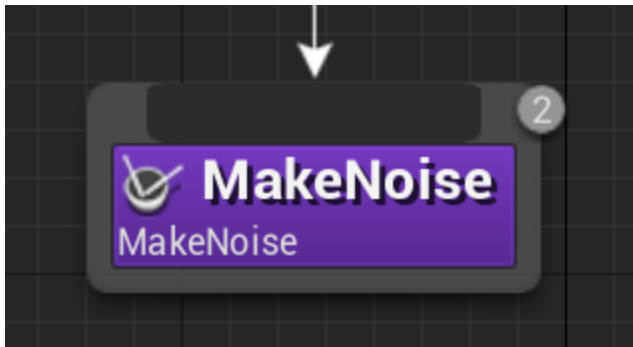
## Finish With Result



The **Finish With Result** Task node can be used to instantly finish with a given result. This node can be used to force a branch to exit or continue based on the defined result.

Property	Description
<b>Result</b>	<ul style="list-style-type: none"> <li>• <b>Succeeded:</b> Finishes with success.</li> <li>• <b>Failed:</b> Finishes with failure.</li> <li>• <b>Aborted:</b> Finishes and aborts.</li> <li>• <b>In Progress:</b> Finishes as in-progress.</li> </ul>
<b>Ignore Restart Self</b>	If enabled, Task search will be discarded when this Task is selected to execute but is already running.
<b>Node Name</b>	The name the node should display in the Behavior Tree graph.

## Make Noise



If the controlled Pawn has the **PawnNoiseEmitter** Component, the **Make Noise** Task will cause the Pawn to "produce a noise" (send a message) that other Pawns with the **PawnSensing** Component can hear (receive the message).

Property	Description
<b>Loudness</b>	How loud the generated sound is.
<b>Ignore Restart Self</b>	If enabled, Task search will be discarded when this Task is selected to execute but is already running.
<b>Node Name</b>	The name the node should display in the Behavior Tree graph.

# Move Directly Toward

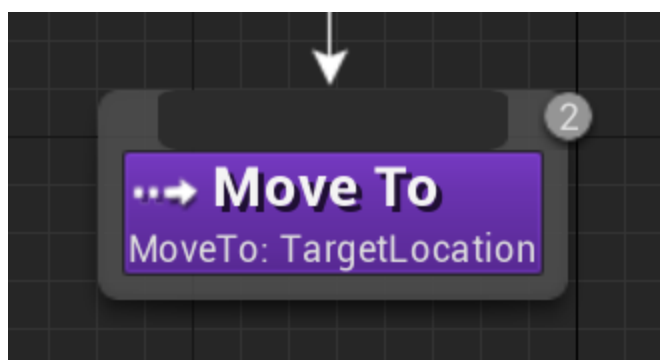


The **Move Directly Toward** Task node moves the AI Pawn toward the specified Actor or Location (Vector) Blackboard entry in a straight line, without regard to any navigation system. If you need the AI to navigate, use the **Move To** Task node instead.

Property	Description
<b>Acceptable Radius</b>	How close does the Pawn have to be to the target for the Task to succeed.
<b>Filter Class</b>	Which navigation data should be used? If set to <b>None</b> the default navigation data will be used.
<b>Allow Strafe</b>	Whether to enable the AI's ability to strafe while moving towards the destination.
<b>Reach Test Includes Agent Radius</b>	If enabled, the radius of the AI's capsule will be added to the threshold between the AI and the goal location.
<b>**Reach Test Includes Goal Radius**</b>	If enabled, the radius of the goal's capsule will be added to the threshold between the AI and the goal location.
<b>Allow Partial Path</b>	If enabled, allow the AI to use an incomplete path when the goal cannot be reached.
<b>Track Moving Goal</b>	If enabled, the path to the goal Actor will update itself when the Actor moves.

Property	Description
<b>Project Goal Location</b>	If enabled, the goal location will be projected on the Nav Mesh before using.
<b>Observe Blackboard Value</b>	If the move goal in Blackboard changes, the move will be redirected to the new location.
<b>Blackboard Key</b>	The Key to check. This is most useful for data types that can return <code>None</code> such as Objects, as other types may return their initialized values (for example 0, false, {0,0,0}).
<b>Ignore Restart Self</b>	If enabled, Task search will be discarded when this Task is selected to execute but is already running.
<b>Node Name</b>	The name the node should display in the Behavior Tree graph.

## Move To

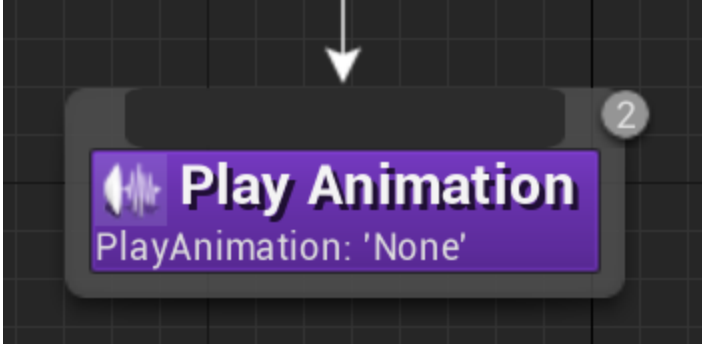


A **Move To** Task will cause a Pawn with a Character Movement component to move using the NavMesh to a Vector Blackboard key.

Property	Description
<b>Acceptable Radius</b>	How close does the Pawn have to be to the target for the Task to succeed.

Property	Description
<b>Filter Class</b>	Which navigation data should be used? If set to <b>None</b> the default navigation data will be used.
<b>Allow Strafe</b>	Whether to enable the AI's ability to strafe while moving towards the destination.
<b>Reach Test Includes Agent Radius</b>	If enabled, the radius of the AI's capsule will be added to the threshold between the AI and the goal location.
<b>Reach Test Includes Goal Radius</b>	If enabled, the radius of the goal's capsule will be added to the threshold between the AI and the goal location.
<b>Allow Partial Path</b>	If enabled, allow the AI to use an incomplete path when the goal cannot be reached.
<b>Track Moving Goal</b>	If enabled, the path to the goal Actor will update itself when the Actor moves.
<b>Project Goal Location</b>	If enabled, the goal location will be projected on the Nav Mesh before using.
<b>Observe Blackboard Value</b>	If the move goal in Blackboard changes, the move will be redirected to the new location.
<b>Blackboard Key</b>	The Key to check. This is most useful for data types that can return <span>None</span> such as Objects, as other types may return their initialized values (for example 0, false, {0,0,0}).
<b>Ignore Restart Self</b>	If enabled, Task search will be discarded when this Task is selected to execute but is already running.
<b>Node Name</b>	The name the node should display in the Behavior Tree graph.

# Play Animation



The **Play Animation** node can be used to play the specified animation asset.

 The animation selected must match the Skeleton of the Pawn the Behavior Tree is controlling.

Property	Description
<b>Animation to Play</b>	Animation asset to play.
<b>Looping</b>	If enabled, the animation will continually loop playback.
<b>Non Blocking</b>	If enabled, the Task will trigger the animation and instantly finish.
<b>Ignore Restart Self</b>	If enabled, Task search will be discarded when this Task is selected to execute but is already running.
<b>Node Name</b>	The name the node should display in the Behavior Tree graph.

# Play Sound



The **Play Sound** node will play the sound given in the **Sound to Play** property.

Property	Description
<b>Sound to Play</b>	The Sound Cue asset to play.
<b>Ignore Restart Self</b>	If enabled, Task search will be discarded when this Task is selected to execute but is already running.
<b>Node Name</b>	The name the node should display in the Behavior Tree graph.

# Push Pawn Action

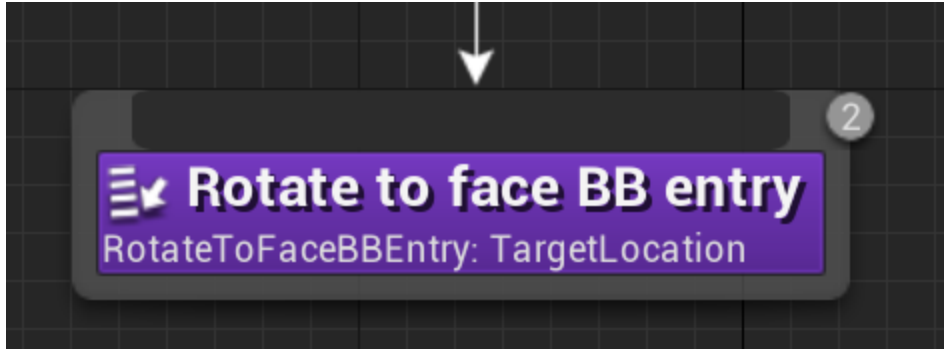


The **Push Pawn Action** node enables you to push the specified action to the Pawn's Controller.

Property	Description
<b>Action</b>	The type of action to push to the Pawn's Controller.
<b>Ignore Restart Self</b>	If enabled, Task search will be discarded when this Task is selected to execute but is already running.

Property	Description
<b>Node Name</b>	The name the node should display in the Behavior Tree graph.

## Rotate to face BB entry



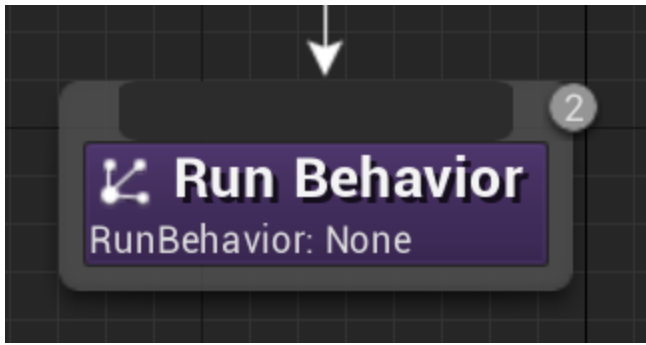
The **Rotate to face BB entry** Task causes the associated Pawn to rotate towards the specified Blackboard Key.

 The Pawn must have **Use Controller Rotation Yaw** enabled to successfully rotate.

Property	Description
<b>Precision</b>	The amount in degrees to be considered as a success condition.
<b>Blackboard Key</b>	The Blackboard Key to get the target to turn toward. This can be either a Vector, Object, or Actor.
<b>Ignore Restart Self</b>	If enabled, Task search will be discarded when this Task is selected to execute but is already running.
<b>Node Name</b>	The name the node should display in the Behavior Tree graph.

## Run Behavior





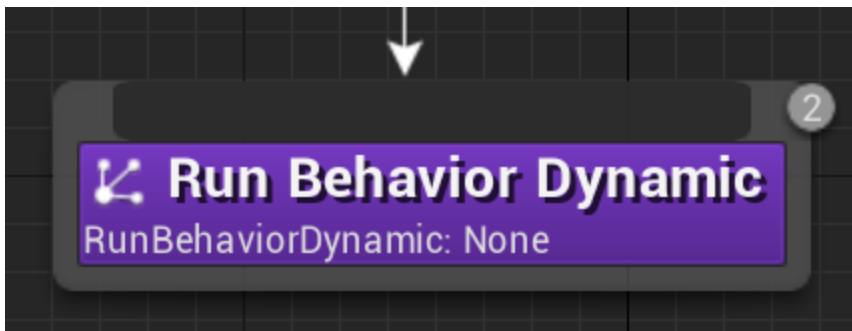
The **Run Behavior** Task enables you to run another Behavior Tree by pushing sub-trees onto the execution stack. One limitation to consider however is that the subtree asset cannot be changed during runtime. This limitation is caused by support for the subtree's Root-level Decorators, which are injected into the Parent tree. Also, the structure of the running tree cannot be modified at runtime.




You can use **Run Behavior Tree Dynamic** if you need a subtree that can be changed at runtime.

Property	Description
<b>Behavior Asset</b>	The <b>Behavior Tree</b> asset to run.
<b>Ignore Restart Self</b>	If enabled, Task search will be discarded when this Task is selected to execute but is already running.
<b>Node Name</b>	The name the node should display in the Behavior Tree graph.

## Run Behavior Dynamic

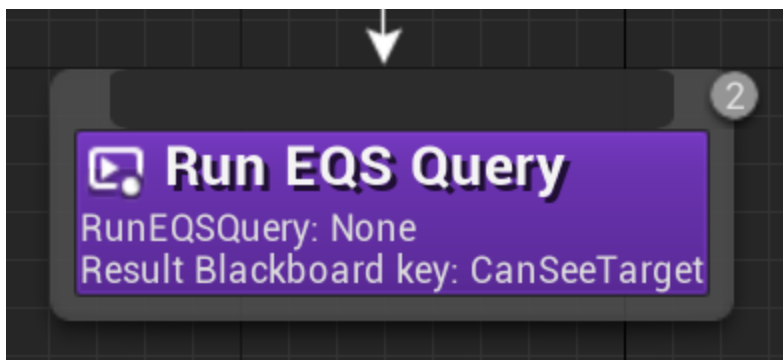


The **Run Behavior Dynamic** Task enables pushing subtrees on the execution stack. Subtree Asset can be assigned at runtime with the **SetDynamicSubtree** function on a **Behavior Tree Component**.

 This does not support a subtree's Root-level Decorator.

Property	Description
<b>Injection Tag</b>	Opens the <b>Gameplay Tag</b> Editor that you can use to identify this Task for subtree injection.
<b>Default Behavior Asset</b>	The initial <b>Behavior Tree</b> asset to run.
<b>Ignore Restart Self</b>	If enabled, Task search will be discarded when this Task is selected to execute but is already running.
<b>Node Name</b>	The name the node should display in the Behavior Tree graph.

## Run EQS Query



The **Run EQS Query** node runs the specified [Environment Query System \(EQS\)](#) asset when the Task node is executed.

Property	Description
<b>Query Template</b>	The EQS asset to run.

Property	Description
<b>Query Config</b>	The additional parameters to include as part of the EQS test.
<b>EQSQuery Blackboard Key</b>	Optional Blackboard Key storing an EQS Query Template to use instead of specifying one under the <b>Query Template</b> .
<b>Run Mode</b>	<ul style="list-style-type: none"> <li>• <b>Single Best Item:</b> Pick the first item with the best score.</li> <li>• <b>Single Random Item from Best 5%:</b> Pick a random item with a score <input type="text" value="95%"/> to <input type="text" value="100%"/> of max.</li> <li>• <b>Single Random Item from Best 25%:</b> Pick a random item with a score <input type="text" value="75%"/> to <input type="text" value="100%"/> of max.</li> <li>• <b>All Matching:</b> Get all items that match conditions.</li> </ul>
<b>Update BBOOn Fail</b>	Updates the Blackboard when the EQS query fails.
<b>Blackboard Key</b>	The Blackboard Key value to update based on the EQS results.
<b>Ignore Restart Self</b>	If enabled, Task search will be discarded when this Task is selected to execute but is already running.
<b>Node Name</b>	The name the node should display in the Behavior Tree graph.

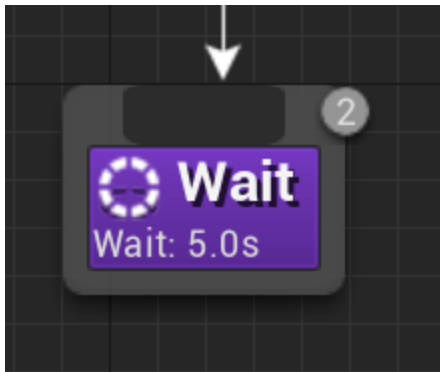
## Set Tag Cooldown



Sets a **Cooldown Tag** value and is used with **Cooldown Tag Decorators** to prevent Behavior Tree execution.

Property	Description
<b>Cooldown Tag</b>	The GameplayTag that will be used for the Cooldown.
<b>Cooldown Duration</b>	How long the Cooldown is in seconds.
<b>Add to Existing Duration</b>	If there is an existing Cooldown on the given GameplayTag, should we add more?
<b>Ignore Restart Self</b>	If enabled, Task search will be discarded when this Task is selected to execute but is already running.
<b>Node Name</b>	The name the node should display in the Behavior Tree graph.

## Wait

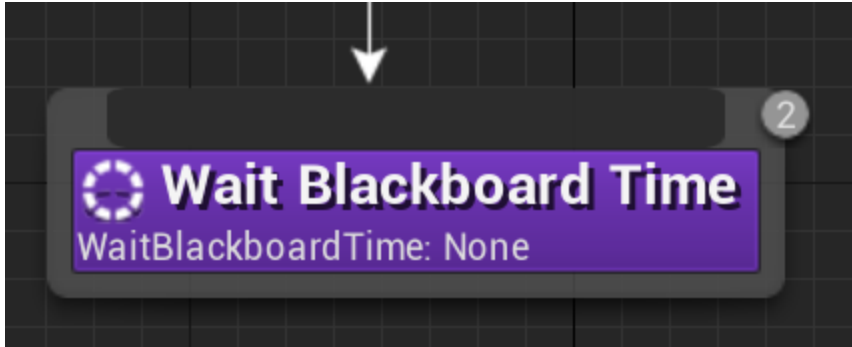


The **Wait** Task can be used in the Behavior Tree to cause the tree to wait on this node until the specified **Wait Time** is complete.

Property	Description
<b>Wait Time</b>	How long to wait, in seconds.
<b>Random Deviation</b>	Allows adding a random time (plus or minus) to the <b>Wait Time</b> property.
<b>Ignore Restart Self</b>	If enabled, Task search will be discarded when this Task is selected to execute but is already running.

Property	Description
<b>Node Name</b>	The name the node should display in the Behavior Tree graph.

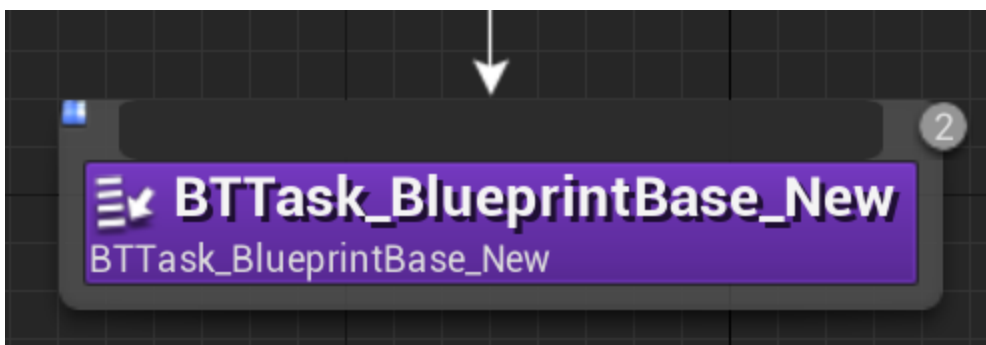
## Wait Blackboard Time



Works just like the **Wait** Task node, except it will pull a Blackboard value for how long it should wait.

Property	Description
<b>Blackboard Key</b>	The float Blackboard Key to reference how long to wait.
<b>Ignore Restart Self</b>	If enabled, Task search will be discarded when this Task is selected to execute but is already running.
<b>Node Name</b>	The name the node should display in the Behavior Tree graph.

## Custom Tasks



You can create new **Tasks** with your custom Blueprint logic and (or) parameters by clicking the **New Task** button.



Blueprint Tasks are slower than native Tasks. If you are looking to optimize your content, switch to using native Tasks.

The following parameters will also be included with your custom logic.

Property	Description
<b>Interval</b>	Defines the time span between subsequent ticks of the Task.
<b>Ignore Restart Self</b>	If enabled, Task search will be discarded when this Task is selected to execute but is already running.
<b>Show Property Details</b>	Show detailed information about properties on the node.
<b>Node Name</b>	The name the node should display in the Behavior Tree graph.