

Thank you for your purchase! The most recent documentation can be found [online](#). If you have any questions feel free to post on the [forums](#) or email [support@opsive.com](mailto:support@opsive.com).

### Column



Arrange the group in one or more columns where the column is significantly longer than the width of rows.

*targetTransform*

The target destination

*targetPosition*

The target destination Vector3 position. Used if targetTransform is null

*lookAhead*

The distance to look ahead to the destination. The higher the value the better the agent will avoid obstacles and less keep formation

*fullSpeed*

The agent move speed after the group has formed

*formationSpeed*

The agent move speed as the group is forming

*waitTime*

The amount of time to wait until the group starts forming

*isLeader*

Should the current agent lead the group? If false the closest agent to the destination will lead

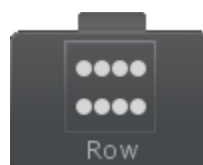
*separation*

The separation between agents

*columns*

The number of columns to form

### Row



Arrange the group in one or more rows with the row significantly wider than the length of the column.

*targetTransform*

The target destination

*targetPosition*

The target destination Vector3 position. Used if targetTransform is null

*lookAhead*

The distance to look ahead to the destination. The higher the value the better the agent will avoid obstacles and less keep formation

*fullSpeed*

The agent move speed after the group has formed

*formationSpeed*

The agent move speed as the group is forming

*waitTime*

The amount of time to wait until the group starts forming

*isLeader*

Should the current agent lead the group? If false the closest agent to the destination will lead

*separation*

The separation between agents

*rows*

The number of rows to form

### Grid



Arrange the group in a grid where the number of rows is equal to the number of columns.

*targetTransform*

The target destination

*targetPosition*

The target destination Vector3 position. Used if targetTransform is null

*lookAhead*

The distance to look ahead to the destination. The higher the value the better the agent will avoid obstacles and less keep formation

*fullSpeed*

The agent move speed after the group has formed

*formationSpeed*

The agent move speed as the group is forming

*waitTime*

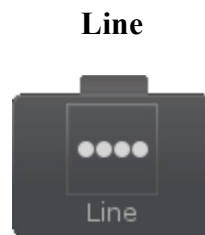
The amount of time to wait until the group starts forming

*isLeader*

Should the current agent lead the group? If false the closest agent to the destination will lead

*separation*

The separation between agents



Arrange the group in a straight horizontal line.

*targetTransform*

The target destination

*targetPosition*

The target destination Vector3 position. Used if targetTransform is null

*lookAhead*

The distance to look ahead to the destination. The higher the value the better the agent will avoid obstacles and less keep formation

*fullSpeed*

The agent move speed after the group has formed

*formationSpeed*

The agent move speed as the group is forming

*waitTime*

The amount of time to wait until the group starts forming

*isLeader*

Should the current agent lead the group? If false the closest agent to the destination will lead

*separation*

The separation between agents

*right*

Should the formation be to the right of the leader?

## Echelon



Arrange the group in a diagonal formation.

*targetTransform*

The target destination

*targetPosition*

The target destination Vector3 position. Used if targetTransform is null

*lookAhead*

The distance to look ahead to the destination. The higher the value the better the agent will avoid obstacles and less keep formation

*fullSpeed*

The agent move speed after the group has formed

*formationSpeed*

The agent move speed as the group is forming

*waitTime*

The amount of time to wait until the group starts forming

*isLeader*

Should the current agent lead the group? If false the closest agent to the destination will lead

*separation*

The separation between agents

*right*

Should the formation be to the right of the leader?

## Wedge



Arrange the group in an upside down V shape where the leader is in the front.

*targetTransform*

The target destination

*targetPosition*

The target destination Vector3 position. Used if targetTransform is null

#### *lookAhead*

The distance to look ahead to the destination. The higher the value the better the agent will avoid obstacles and less keep formation

#### *fullSpeed*

The agent move speed after the group has formed

#### *formationSpeed*

The agent move speed as the group is forming

#### *waitTime*

The amount of time to wait until the group starts forming

#### *isLeader*

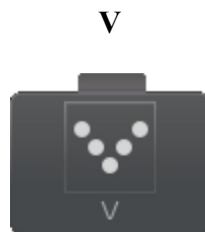
Should the current agent lead the group? If false the closest agent to the destination will lead

#### *separation*

The separation between agents

#### *fill*

Should agents fill the wedge?



Arrange the group in a V shape where the leader is in the back.

#### *targetTransform*

The target destination

#### *targetPosition*

The target destination Vector3 position. Used if targetTransform is null

#### *lookAhead*

The distance to look ahead to the destination. The higher the value the better the agent will avoid obstacles and less keep formation

#### *fullSpeed*

The agent move speed after the group has formed

#### *formationSpeed*

The agent move speed as the group is forming

#### *waitTime*

The amount of time to wait until the group starts forming

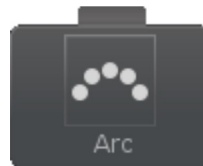
*isLeader*

Should the current agent lead the group? If false the closest agent to the destination will lead

*separation*

The separation between agents

### Arc



Arrange the group in an arc.

*targetTransform*

The target destination

*targetPosition*

The target destination Vector3 position. Used if targetTransform is null

*lookAhead*

The distance to look ahead to the destination. The higher the value the better the agent will avoid obstacles and less keep formation

*fullSpeed*

The agent move speed after the group has formed

*formationSpeed*

The agent move speed as the group is forming

*waitTime*

The amount of time to wait until the group starts forming

*isLeader*

Should the current agent lead the group? If false the closest agent to the destination will lead

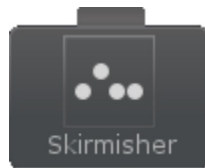
*radius*

The radius of the arc

*concave*

Is the arc concave?

### Skirmisher



Arrange the group in a randomly spread out line.

*targetTransform*

The target destination

*targetPosition*

The target destination Vector3 position. Used if targetTransform is null

*lookAhead*

The distance to look ahead to the destination. The higher the value the better the agent will avoid obstacles and less keep formation

*fullSpeed*

The agent move speed after the group has formed

*formationSpeed*

The agent move speed as the group is forming

*waitTime*

The amount of time to wait until the group starts forming

*isLeader*

Should the current agent lead the group? If false the closest agent to the destination will lead

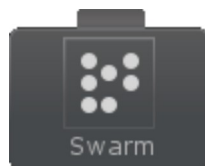
*minSeparation*

The minimum separation between two agents

*maxSeparation*

The maximum separation between two agents

## Swarm



Arrange the group in a tight circle that can move together.

*targetTransform*

The target destination

*targetPosition*

The target destination Vector3 position. Used if targetTransform is null

*lookAhead*

The distance to look ahead to the destination. The higher the value the better the agent will avoid obstacles and less keep formation

*fullSpeed*

The agent move speed after the group has formed

*formationSpeed*

The agent move speed as the group is forming

*waitTime*

The amount of time to wait until the group starts forming

*isLeader*

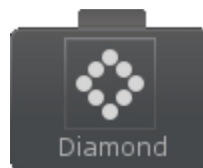
Should the current agent lead the group? If false the closest agent to the destination will lead

*radius*

The radius of the group

*maxPlacementAttempts*

The agent positions in the swarm is randomly determined. These positions will keep being regenerated until they are not overlapping any other position. This value allows the random placement to be capped so it doesn't result in an infinite loop

**Diamond**

Arrange the group in a tactical diamond shape.

*targetTransform*

The target destination

*targetPosition*

The target destination Vector3 position. Used if targetTransform is null

*lookAhead*

The distance to look ahead to the destination. The higher the value the better the agent will avoid obstacles and less keep formation

*fullSpeed*

The agent move speed after the group has formed

*formationSpeed*

The agent move speed as the group is forming

*waitTime*



The amount of time to wait until the group starts forming

*isLeader*

Should the current agent lead the group? If false the closest agent to the destination will lead

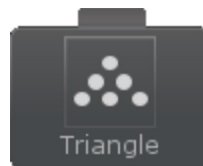
*separation*

The separation between agents

*backPositionOffset*

Should the back agents have a left and right offset?

### Triangle



Arrange the group in a triangle.

*targetTransform*

The target destination

*targetPosition*

The target destination Vector3 position. Used if targetTransform is null

*lookAhead*

The distance to look ahead to the destination. The higher the value the better the agent will avoid obstacles and less keep formation

*fullSpeed*

The agent move speed after the group has formed

*formationSpeed*

The agent move speed as the group is forming

*waitTime*

The amount of time to wait until the group starts forming

*isLeader*

Should the current agent lead the group? If false the closest agent to the destination will lead

*length*

The length of the triangle

### Square



Arrange the group in a square.

*targetTransform*

The target destination

*targetPosition*

The target destination Vector3 position. Used if targetTransform is null

*lookAhead*

The distance to look ahead to the destination. The higher the value the better the agent will avoid obstacles and less keep formation

*fullSpeed*

The agent move speed after the group has formed

*formationSpeed*

The agent move speed as the group is forming

*waitTime*

The amount of time to wait until the group starts forming

*isLeader*

Should the current agent lead the group? If false the closest agent to the destination will lead

*length*

The length of the square

## Circle



Arrange the group in a circle.

*targetTransform*

The target destination

*targetPosition*

The target destination Vector3 position. Used if targetTransform is null

*lookAhead*

The distance to look ahead to the destination. The higher the value the better the agent will avoid obstacles

and less keep formation

### *fullSpeed*

The agent move speed after the group has formed

### *formationSpeed*

The agent move speed as the group is forming

### *waitTime*

The amount of time to wait until the group starts forming

### *isLeader*

Should the current agent lead the group? If false the closest agent to the destination will lead

### *radius*

The radius of the circle

## Formation Type Creation

As we hear more suggestions from the community the number of formation tasks will continue to grow. However, you may want to add your own custom formation and this can easily be accomplished with the Fomration Pack API. The first step in developing your own formation task is to inherit your task from the GroupFormation base class. This will take care of most of the work for you and all you have to do is specify where you want each of your units to be positioned. The following methods can be overridden:

```
/// Virtual method to allow the formation tasks to specify a target position.
/// The index of the group member.
/// The z distance to look ahead of the target position.
/// The position to move to, in world space.
protected Vector3 TargetPosition(int index, float zLookAhead)

/// Adds the agent to the formation group.
/// The agent to add.
/// The index of the agent within the group.
protected void AddAgentToGroup(Behavior agent, int index)

/// Removes the agent from the group.
/// The agent to remove.
/// The index of the agent removed from the group.
protected int RemoveAgentFromGroup(Behavior agent)
```

TargetPosition is the method that will position your agent. The index parameter specifies the index of the agent within the formation. zLookAhead specifies the distance to look ahead of the target position. As a basic example, the following method will form a line where each agent is two units apart on the relative x axis:

```
protected override Vector3 TargetPosition(int index, float zLookAhead)
{
    return transforms[0].TransformPoint(index * 2, 0, zLookAhead);
}
```

The transforms array is a protected array belonging to the parent class. It contains a list of all of the group transforms. In this example we are getting the relative offset from the first transform, which is always the leader transform. As the index increases the distance between the agent and the leader will also increase on the x axis which will form a line. zLookAhead is used on the relative z axis to prevent the agent from always

being within stopping distance from its target.

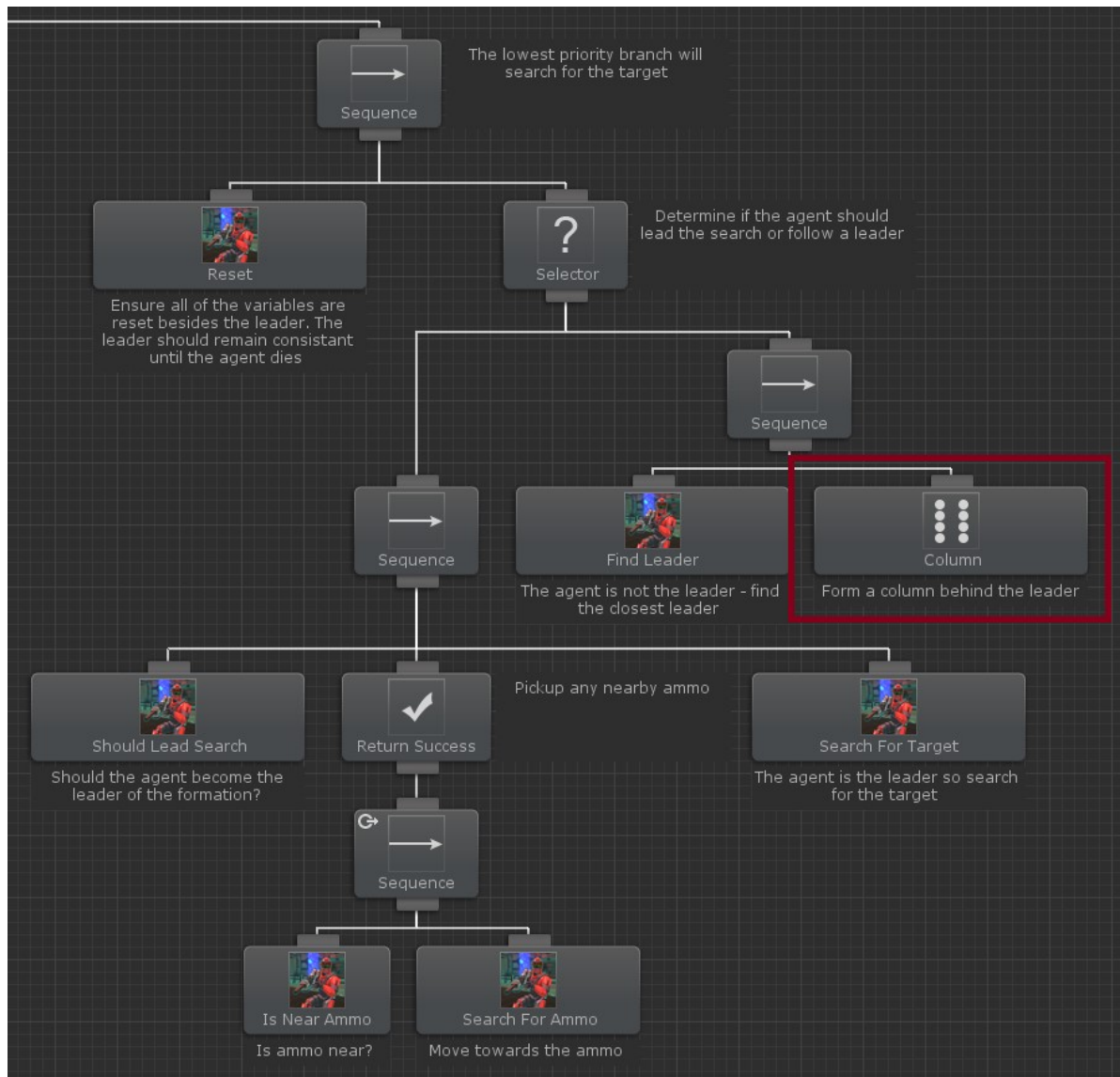
AddAgentToGroup and RemoveAgentFromGroup are called when the agent is added or removed from the group. This allows for initialization/destruction if the formation task requires knowledge ahead of time of how many agents are in the group.

### Third Person Controller Integration

As of version 1.2.1 of the [Third Person Controller](#) no separate integration is required - all of the components are included in the [Third Person Controller package](#).

### Deathmatch AI Kit Integration

The Formations Pack is integrated with the [Deathmatch AI Kit](#) which allows your deathmatch agents to use a formation other than the deathmatch V Formation task. Using the Formations Pack within the deathmatch tree can be done by swapping out the V Formation with any of the Formations Pack tasks. For example, the image below shows the Column task being used in the last branch of the Team tree:



More information on how the team tree is structured can be found on [this page](#).

### **Apex Path Integration**

The Formations Pack tasks are integrated with [Apex Path](#). The Apex Path files are located on the [integrations page](#) because the Tactical Pack does not require Apex Path to work. To use the Apex Path tasks, you must first add all of the Apex Path components to your agent. This can be added via the Components -> Apex -> QuickStarts -> Navigating Unit menu option.

### **Support**

We are here to help! If you have any questions/problems/suggestions please don't hesitate to ask. You can email us at [support@opsive.com](mailto:support@opsive.com) or post on the [forum](#).