



APEXPATH

COMPONENTS

Contents

Version History.....	3
Introduction	4
Common	5
Advanced Message Bus	5
Game Services Initializer	5
Load Balancer.....	5
Debugging	7
Grid Visualizer	7
Path Visualizer.....	8
Steering Monitor.....	8
Input	9
Selection Rectangle.....	9
Very basic input receiver	9
Navigation.....	10
Basics	10
Grid	10
Layer Mapping	11
Path Service.....	12
Terrain Height Map	12
Behaviours	13
Patrol.....	13
Wander	13
Obstacles	14
Dynamic Obstacle	14
Invalid Destination	14
Props	15
Patrol Point	15
Patrol Route.....	15
Steering	16
Humanoid Speed	16
Steerable Unit	16
Basic Scanner	17
Steer for Basic Avoidance.....	17
Steer to Flee Obstacle	17
Steer for Path.....	18

Replan when Blocked	19
Replan when No Route	19
Turnable Unit	20
Portals.....	21
Grid Portal.....	21
Portal Action Teleport	22
QuickStarts.....	23
Game World	23
Game World with Basic Height Map.....	23
Game World with Terrain Height Map	23
Navigating Unit	23
Navigating Unit On patrol.....	23
Navigating Unit with Selection.....	23
Navigating Unit Wandering	23
Patrol Route.....	23
Steering: Basic Avoidance	23
Triggers	24
Attribute Manipulating Trigger	24
Units	25
Unit	25
Grid Aware Unit	25
Selectable Unit	25

Version History

Version 1.0	Initial Version
Version 1.1	Portals added
Version 1.2	Added new avoidance steering
Version 1.3	A few new properties.
Version 1.4	Update for Apex Path 1.2

Introduction

Apex Path offers a wide range of components to control how units move around in the game world, and how they interact with obstacles and other entities.

This document provides a detailed description of each of these components.

For a full reference on all types in the Apex Path framework, please refer to the Apex Path API help file.

Common

Advanced Message Bus

Apex Path uses a Basic Message Bus as its default. To use the advanced alternative instead you can add this component to your scene's game world.

In most cases there is no reason to change the default.

Please refer to the Apex Path Extensibility guide for details on message buses and the difference between the basic and advanced version.

Game Services Initializer

This is a required component that must always reside on the Game World GameObject. It initializes core services used by other components.

As with most other components, it is automatically added to the scene when using the Quick Starts.

Load Balancer

This component holds the configuration of each of the load balancer queues used by Apex Path.

Property	Description
Delayed Actions	The load balancer used for executing delayed actions. This queue is also available for use to you. Please refer to the Apex Path Extensibility guide for more on load balancers and their use.
Steering	The load balancer that handles steering
Dynamic Obstacles	The load balancer that handles dynamic obstacles

Each load balancer has the following configuration options

Property	Description
Update Interval	This is the interval in seconds between each update of items in the queue.
Max Updates Per Frame	This defines the maximum number of items in the queue that are updated each frame. If more items than this are ready to be updated (their interval has passed) they will instead be updated on the next frame. If you have a large number of units in the scene you may need to increase the default value, to ensure that all units get processed as close to their intended interval as possible.

Max Update Time in Milliseconds Per Interval

This defines the maximum number of milliseconds the load balancer is allowed to use each frame. If it reaches this limit it will continue processing the remaining items on the next frame.

This property is related to the above property, as they both control how much time the load balancer is allowed to use each frame.

As with the above property you may need to increase the default value if you have a large number of units, and start experiencing erratic behavior.

Debugging

Components in this section are there to provide visual clues and references by drawing Gizmos or outputting text to the console.

Grid Visualizer

Draws a visual representation of grids attached to the same GameObject. The grid also shows which nodes are blocked by obstacles.

Property	Description
Draw Always	Setting this to true means that the grid is drawn even if the GameObject is not selected.
Draw Mode	<p><i>Layout:</i> Draws the actual representation of the grid. It is useful as a reference when placing geometry in the scene.</p> <p><i>Accessibility:</i> Draws the grid as accessibility lines, meaning that all lines drawn indicate a valid path to move through the level (not counting Dynamic Obstacles). If the scene has a height map, the grid(s) must be baked for height map data to be represented.</p>
Draw Sub Sections	Draws the subsections of the Grid (See Grid component for details).
Draw All Grids	Controls whether the visualizer draws all grids in the scene or only those on the same GameObject as the visualizer
Editor Refresh Delay	The delay following updates to the scene after which the editor refreshes the grid while in editor mode.
Draw Distance Threshold	<p>This controls how much of the grid to draw, which in effect means that of the camera is tilted or zoomed out a certain amount the grid will no longer be drawn. This is both to improve performance, but also to avoid drawing visualizations that as useless, e.g. too far away.</p> <p>The threshold used is the diagonal distance of the portion of the grid currently visible on screen.</p>
Grid lines color	Sets the color of the grid lines drawn.
Obstacle color	Sets the color of the obstacles markings in the grid.
Sub Sections color	Sets the color of the lines marking the subsections' boundaries.

Path Visualizer

Draws a line outlining the path the unit is about to travel. Add this component to a unit if you need to visually debug the path the unit receives. Remember to enable gizmos in the Game View to see the path.

Property	Description
Draw Always	Draws the path disregarding whether the unit is selected in the Hierarchy.
Route Color	Sets the color of the line outlining the route the unit is about to travel
Waypoints Color	Sets the color of waypoints along the route
Show Segment Markers	Toggles whether an indication of the individual path segments is drawn or not.

Steering Monitor

Writes steering events to the console for debug purposes. For instance when a unit reaches a waypoint or reaches its destination.

Property	Description
Unit Filter	Can be used to only show events for a specific unit. Draw the unit's GameObject to the property to assign the filter.
Event Filter	Can be used to filter what events the monitor reacts to.

Input

Selection Rectangle

This component is used to create the selection rectangle that appears when you left click and drag in game mode, to select units.

It is used by the Selection Rectangle prefab, which is automatically added to the game world when you create one or more selectable units using the Quick Starts.

Property	Description
Start Delta Threshold	Controls how far you have to drag the mouse after holding down the left mouse button (or other input you may define), before the rectangle starts drawing.

Very basic input receiver

As the name suggests, this is a rather naïve implementation of an input receiver. All input keys and buttons are hard coded.

It works fine for getting started, but it is strongly advised that you create your own input receiver at some point.

Please refer to the Apex Path Extensibility guide for details on doing so.

Navigation

Basics

Grid

The grid component holds the information about the cells used by the pathfinder and other components.

Sub sections in the grid are used for dynamic re-planning to improve performance. Whenever an obstacle changes position, only units in the sub sections of the grid will be notified.

Property	Description
Friendly Name	An optional name of the grid.
Link Origin to Transform	If this is checked, the origin (center) of the grid is the same as the position of the transform.
Origin	Center of the grid
Size X	The size of the grid on the X-axis specified in number of cells.
Size Z	The size of the grid on the Z-axis specified in number of cells.
Sub Sections X	How many subsections the grid is divided into on the X-axis.
Sub Section Z	How many subsections the grid is divided into on the Z-axis.
Sub Sections Cell Overlap	The amount cells sub sections overlap
Cell Size	The size of each cell in the grid. Cells size should be large enough to hold the largest unit type.
Lower Boundary	How far below the origin to sample heights. Any geometry lower than this will be ignored.
Upper Boundary	How high above the origin to sample heights. Any geometry higher than this will be ignored.
Height Granularity	The distance between each height plot point. The lower this number the more accurate the height map will be, at the cost of some memory.
Obstacle Sensitivity Range	<p>Sets the range from the center of the cell, such that any obstacle inside this range will block the cell.</p> <p>This should be set to the radius of the largest unit you plan to navigate the grid. (See Unit.Radius)</p>

Max Walkable Slope Angle	Controls how steep a slope can be before it is deemed unwalkable.
Max Scale Height	<p>Controls the maximum height that is considered scalable (i.e. units are able to walk over) even if the slope is higher than the above setting.</p> <p>This is useful for stairs and such, and in general should be set to a value other than 0. If set to zero, even the smallest height difference will be considered unwalkable.</p>
Automatic Initialization	<p>When checked (default) the grid will initialize itself when enabled.</p> <p>If unchecked you will need to call the Initialize method of the grid component to enable and initialize it. This is useful for dynamic instantiation of grids at runtime, since the manual initialization is load balanced.</p>
Store Grid data as Asset	If checked, baking the grid will store the data in a separate asset, which enables reuse, i.e. prefab'ing the grid.
Bake Grid	<p>Pressing this button will create a cached version of the grid. Having a cached version of the grid reduces start-up time since the grid is already calculated.</p> <p>Some visualization also require the grid to be baked; again for performance reasons.</p>

Layer Mapping

This component maps the predefined layers of Apex path to layers in the project.

Property	Description
Static Obstacle Layer	The layer(s) that holds static obstacles, i.e. obstacles that will not change during the course of the game. This mapping is optional.
Terrain Layer	The layer(s) that holds the ground elements on which units can move.
Unit Layer	The layer(s) assigned to units. This mapping is optional.

Path Service

This is the service responsible for supplies each unit with a valid path.

The path service should be added to the scene only once.

Property	Description
Engine Type	<p>Here you can select the pathfinding algorithm to use.</p> <p>Astar: Classic pathfinding algorithm, guaranteed to find a path.</p> <p>JumpPointSearch: Memory optimized A* algorithm up to x5 faster in most cases.</p>
Initial Heap Size	<p>The heap size should be sized after the grid. A ballpark figure is number of nodes in the grid divided by 10. A 10x10 grid should have an initial heap size of 10.</p> <p>You are not required to set this as the heap will automatically resize itself as needed, but for maximum performance it is better to have a fitting start size to avoid automatic resizing during runtime.</p>
Run Async	When this is checked the pathfinding will use multithreading
Use Thread Pool for Asyn Operations	<p>Controls whether the path engine uses a thread pool or a dedicated thread for async mode.</p> <p>Even if set to use a thread pool, it will only do so if the platform supports it.</p>
Max milliseconds per frame	The maximum number of milliseconds that will be use per frame when the pathfinder is not running in Async mode.

Terrain Height Map

Adds height map navigation using a Unity3D Terrain component.

This component is automatically added to the scene at runtime, if any terrain components are detected.

Property	Description
Terrain	The Terrain component reference.
Update Grids	Clicking this will update any prebaked grids with changes made to the height map.

Behaviours

Patrol

The Patrol Behavior is a simple behavior that allows the Agent to move along a predefined route.

Property	Description
Route	The Patrol Route Component linked to the Patrol Behaviour (see Patrol Route). This is the patrol route the Agent uses when patrolling.
Linger At Nodes for Seconds	The amount of time an Agent rests when reaching a point on the Patrol Route, before the Agent starts moving towards the next point.
Reverse Route	If set to true the Agent traverses the Patrol Route in order from the last node to the first.
Randomize	Instead of following the route in order the unit will randomly choose the next point on the route to go to.

Wander

The Wander Behavior makes the Agent wander at random across the Grid. This behavior is useful for debugging a level or for random characters in the scene such as critters.

Property	Description
Radius	The maximum radius from the Agent's original position that the next destination is sampled from.
Minimum Distance	The minimum radius from the Agent's position that the next destination is sampled from, e.g. the minimum distance you want to move each time the agent start to wander in a new direction.
Linger for Seconds	The amount of time an Agent rests when reaching its destination, before the Agent starts moving towards the next destination.
Bail After Failed Attempts	This is a failsafe to ensure that in case everything around the Agent is blocked or otherwise inaccessible, he will not continue to attempt to find a new route indefinitely.

Obstacles

Dynamic Obstacle

This component is for obstacles that are dynamic in nature, i.e. they change status, size or position during the course of the game.

Property	Description
Exceptions	The obstacle will not block units with these attributes
Update Mode	Controls when the component updates the grid. <i>AtInterval</i> : The obstacle updates every set interval. <i>OnRequest</i> : The obstacle updates only if requested to do so, through scripting.
Velocity Prediction Factor	Indicates how far ahead of the obstacles moving direction (velocity) the obstacle will block the grid.
Stop Updating If Stationary	If set to true and the obstacle is not moving, it will remove itself from the update queue.
Stationary Threshold Seconds	Indicates the number of seconds the obstacle will have to be stationary in order to stop updating. Only valid if "Stop Updating If Stationary" is set to true.
Custom Update Interval	Setting this to a number overrides the default update interval specified on the dynamic obstacles load balancer.

Invalid Destination

Marks a transform or sub area of a transform as an invalid destination for path finding. This means that while the area is traversable it cannot be a final destination.

Property	Description
Entire Transform	Check this to make the entire transform invalid as a destination.
Only Sub Area	Use this to specify a specific area as invalid as a destination. Only used if the 'Entire Transform' property is set to false.

Props

Patrol Point

A patrol point represents a single waypoint along a patrol route. Patrol points are typically added as child GameObjects of a Patrol Route.

Property	Description
Order Index	An arbitrary index used to order patrol points in relation to each other.
Use Transform Position	The location of the patrol point is that of its transform.
Location	If 'Use Transform Position' is set to false, this location is used to position the patrol point instead.

Patrol Route

Patrol route is used by the patrol behavior and is the parent of two or more patrol points. The patrol routes can be reused across units allowing units to share patrol routes.

The units will patrol along the patrol points in the specified order.

The recommended way of adding a patrol is to use the quick start components: Components -> Apex -> Quick Starts -> Patrol Route.

Property	Description
Draw Gizmos Always	Setting this to true means that the patrol route is drawn even if the GameObject is not selected.
Gizmo Color	The color of the patrol points on this route..

Steering

Humanoid Speed

On this component you set the speed for humanoid characters based on multiple modes of movement. The component sets the maximum speeds for different move modes for the unit. It's possible to create other components for different types of creatures.

This component starts out in Walk mode. In order to change movement mode you will have to call the appropriate method on the component through scripting. See the examples project for... yes an example.

Property	Description
Minimum speed threshold	The speed under which the unit will stop moving to avoid jittering
Crawl	The maximum speed of the unit when crawling
Crouched Speed	The maximum speed the unit when the unit is in a crouching position
Walk Speed	The maximum speed of the unit when it is walking
Run Speed	Maximum speed of the unit when it is running forward
Strafe Max Speed	Maximum speed of the unit when it is moving sideways in a strafing position
Back Pedal Max Speed	Maximum speed of the unit when it is moving backwards

Steerable Unit

The component takes input from all steering components on the unit and combines them into a final velocity of the unit.

Property	Description
Current Speed	Show the current speed of the unit in m/s.
Velocity Smoothing Rate	The speed by which velocities change. The higher the number the faster the change will happen.
Turn Request Priority	The priority is used in conjunction with other components that are also determining the orientation of the unit.
Turn Speed	This is maximum speed by which the unit can turn
Minimum Speed to Turn	The minimum speed the unit must be travelling at in order to turn.

Stop If Stuck for Seconds	This is the maximum time a unit can be stuck before it stops.
----------------------------------	---

Basic Scanner

The component provides the perception needed by the basic steering.

Property	Description
Scan Interval	The interval between scans.
Scan Radius	The radius around the unit that will be scanned for other units.

Steer for Basic Avoidance

The component steers the unit in order to avoid other units in its path.

Property	Description
Weight	The weight relative to other steering components

Steer to Flee Obstacle

Steering behaviors that will make the unit avoid obstacles encroaching on its position. If an obstacle suddenly covers the cell in which the unit is located, the unit will try and flee the cell.

Property	Description
Weight	All steering components can specify a weight, which determines how much impact they will have on the final velocity compared to other attached steering components.
Flee Max Radius	The maximum distance in cells that the unit will inspect to find a spot to flee to.

Steer for Path

The component steers the unit along a path received from the path finder.

Property	Description
Weight	The weight relative to other steering components
Pathing priority	The priority that will be given to this unit's path request in relation to other units requesting paths.
Use Path Smoothing	Whether or not to smooth the path received from the path finder. Path smoothing creates more natural move behavior at a very low cost.
Allow Corner Cutting	When this is set to true the unit will select a path very close to corners of obstacles. When set to false the unit will keep at least half the distance of a cell to the corners of obstacles.
Prevent Off Grid Navigation	If this is checked the unit will not be able to move off grid, and all navigation between grids must be done using portals.
Prevent Diagonal Moves	If this is checked, the unit will not make diagonal moves, i.e. it will only move along the axis. This setting is only valid when using the A* algorithm, not Jump Point Search.
Navigate to Nearest if Blocked	If checked, units that are given a destination to which they cannot move (blocked or otherwise inaccessible) will instead move as close to that destination as possible. If unchecked units that receive an invalid destination will simply ignore it, or use the attached result processors to handle it.
Slowing Distance	The distance within which the unit starts to slow down when approaching its final destination
Slowing Algorithm	The Linear algorithm slows the unit at a constant rate. The Logarithmic algorithm slows the unit at a logarithmic rate, subtle at first and increasingly more the closer the unit gets to the end destination.
Request Next Waypoint Distance	The distance from the current waypoint at which the next waypoint will be requested

Proximity Evaluation Min Angle	<p>The angle within which an evaluated point must be in, in order for the unit to consider it.</p> <p>This is used to ensure that the unit does not start to slow down or request a new waypoint unless it is approaching these from a proper angle.</p>
Next Node Distance	<p>The distance from the center of the node on the path the unit is currently moving towards where the unit will start to move on to the next node on the path.</p>
Arrival Distance	<p>The distance from the final destination where the unit will stop. This is to stop jittering when arriving at the final destination.</p>
Announce All Nodes	<p>If checked the unit will announce each time it reaches a node along its path, with the exception of waypoint nodes or the final destination, since they have their own event messages.</p>
Replan Mode	<p><i>At Interval:</i> The unit replans at a set interval</p> <p><i>Dynamic:</i> The unit replans when its surroundings change.</p> <p><i>NoReplan:</i> The unit does not replan at all.</p>
Replan Interval	<p>In AtInterval mode the replan interval is the fixed interval in seconds between replanning.</p> <p>In Dynamic mode the replan interval is the minimum required time between each replan.</p>

Replan when Blocked

This component is a result processor that will change the default behavior in cases where a valid path could not be found due to the destination being blocked.

It will try to find an alternative destination which is as close to the original destination and to the unit as possible.

Property	Description
Processing Order	<p>The order this processor should be processed in relation to other processors.</p>
Max Cell Distance for New Destination	<p>The maximum distance in cells that the unit will inspect to find a new destination.</p>

Replan when No Route

This component is a result processor that will change the default behavior in cases where a valid path could not be found as the destination cannot be reached.

It will try to replan the same route a number of times to see if the cause of the destination being unreachable was temporary.

Property	Description
Processing Order	The order this processor should be processed in relation to other processors.
Retry Delay	The delay in seconds between each retry.
Max Retries	The maximum number of retries.

Turnable Unit

This component turns the unit to face in a direction specified by other components.

For instance the Steerable Component will request to turn the unit so it is facing the direction of movement.

Property	Description
Ignore Axis	<p>If you want the unit to ignore a specific axis component when turning, set this to the appropriate value.</p> <p>Note that it is not the actual axis that is ignored, but the axis component of the Vector3, e.g. setting this to Y, will keep the unit upright and only turn it AROUND the Y axis.</p>

Portals

Portals provide the means to perform special navigation within a grid or between grids, for example a teleport.

While portals add a great many possibilities, they also come with some implications that you need to be aware of before starting to use portals.

1. Adding a Shortcut portal to a grid increases the time used by the path finder to find a route, by 30%. That is in the worst case scenario and **ONLY** applies to Shortcut portals, not Connectors. This increase is per portal, so adding 10 portals increases path finding time by a factor of 3. If the portals are between grids and not on the same grid, the time cost is half, i.e. 15%. Normally however you would use Connector portals between grids, which have no overhead.

Of course portals also have the opposite effect, when a portal acts as a short cut; that also means that the path finder uses far less time.

Since path finding is fast, the added cost is not likely to be an issue unless you go overboard and add hundreds of portals.

2. A* path finding usually guarantees the shortest path to the goal. With portals that is not the case. In short what this means that the path finder will prefer routes that move towards the goal, so any portals that bring the path closer to the goal will be preferred. Due to how portals work this may not be the shortest path, e.g. there may be a shorter path that uses a portal that initially moves away from the goal but eventually turns out to be shorter since it can use another portal to get closer.
3. If a portal is set up between two grids, the normal inter-grid navigation will no longer be active; instead the portal will be used. Note that this is regardless of how far or close the unit is to the other grid. If at least one portal exists between two grids, navigation will always use portals.

Grid Portal

This is the component that defines a portal. It consists of two sections that define the two sides of the portal.

Property	Description
Exclusive To	Controls which unit type that can use the portal. If set to anything other than <i>Nothing</i> the portal will only be usable by units with matching attribute(s).
Portal Name	The name of the portal. This can be used to get a reference to the portal through the Portal Manager (script).

Type	Either Shortcut or Connector. As a general rule use connectors as much as possible and only use Shortcut portals when needed.
Portal One	One side of the portal
Portal Two	The other side of the portal
Portal Color	The color used to draw the portal Gizmo

Portal Action Teleport

This is the basic portal action that ships with Apex Path.

This action simply teleports the unit from one end of the portal to the other.

Please refer to the *Apex Path Extensibility* guide for details on creating your own portal actions.

QuickStarts

Quick Starts are components that act as factories to add various other components to a GameObject.

Most quick starts will also add and setup a Game World object if one does not exist.

Game World

Adds a Game World object to the scene and configures it with the Apex path high level components, such as the Grid and Path service.

Game World with Basic Height Map

Adds and configures a Game World object to the scene the addition of a Basic Height Map.

Game World with Terrain Height Map

Adds and configures a Game World object to the scene the addition of a Terrain Height Map.

Navigating Unit

Adds the components necessary for a unit to have basic navigation skills. This is typically an enemy or another unit that is moved by AI instead of player interaction.

Navigating Unit On patrol

Adds the components necessary to navigate and adds the patrol behavior. This means that the unit will patrol according a patrol route. The patrol route will be created as a child of the GameWorld object.

Navigating Unit with Selection

Adds the components necessary to navigate and adds the ability to select and move a player controllable unit using the mouse or other input device.

Navigating Unit Wandering

Adds the components necessary to navigate and adds the wander behavior.

Patrol Route

Adds a Patrol Route with two Patrol Points

Steering: Basic Avoidance

Adds basic avoidance to a unit, to prevent it from colliding with other units.

Triggers

Attribute Manipulating Trigger

This component removes or adds attributes to units that collide with the GameObject that has this component attached. See the chapter on Attributes in the Apex Path User Manual or Extensibility guide for further information on Attributes.

Property	Description
Applies	Units colliding will have the attribute mask in the dropdown applied to them.
Removes	Units colliding will have the attribute mask in the dropdown removed from them.
Update On	Controls whether units will have the attribute masks applied and/or removed when they enter the collider, exit the collider, or both.

Units

Unit

Holds some basic properties associated with all units.

Property	Description
Attributes	<p>If an Attribute set has been set up, this allows the assignment of attributes to the unit.</p> <p>For details on setting up an Attribute set, please refer to the Apex Path User Manual or Extensibility Guide.</p>
Radius	The radius of the unit.
Field of View	The unit's field of view in degrees, from the left extreme to the right extreme.
Y Axis Offset	<p>If the unit has no rigidbody or if gravity has been disabled, this should be set to the unit's natural offset above the ground.</p> <p>It can be set on units with rigidbodies as well, but will simply be ignored as long as the rigidbody uses gravity.</p> <p>If the center of your unit, when the unit's base is resting on the ground, is 0.5, set this to 0.5 as well. This ensures that the unit stays grounded even when no physics are applied.</p>

Grid Aware Unit

This is a very basic component that gives the unit a reference to the Grid on which it started out.

Selectable Unit

Units with this component attached are selectable using input devices.

Property	Description
Selection Visual	This gameobject represents the visual that is shown around or under the unit when the unit is selected.