

## Release Notes

**Always delete the previous Behavior Designer files before updating! If you store your own files in the Behavior Designer directory make sure you don't delete those as well.**

### Version 1.5.10 Changes:

- Allow abstract types in fields
- Added the OnConditionalAbort task callback which gets called when a task is ended because of a conditional abort
- Fixed the behavior tree from not always being positioned in the center when initially loading
- Fixed null reference when linking a task which doesn't have the same field names
- Fixed null reference when removing a referenced task

---

### Version 1.5.9 Changes:

- Added support for Unity 5.6
- Removed support for Unity 5.0
- Added the Parallel Complete task
- Added the following editor callbacks: BehaviorDesignerWindow.OnAddTask, BehaviorDesignerWindow.OnRemoveTask
- Behavior.OnBehaviorStart/OnBehaviorRestart/OnBehaviorEnd callbacks now include the Behavior component
- Added support for Task.StopCortine(coroutine)
- Added callback for Task.OnAnimatorIK
- Use a platform consistent hash for binary serialization
- Added error checking to the property mapping
- Foldouts will remember their foldout status after serialization
- Fixed variables from not being set correctly on multiple nested trees
- Fixed slowdown when editing a large binary tree on the prefab instance

---

### Version 1.5.8 Changes:

- Added support for Unity 5.5
- Added support for retina screens
- Conditional aborts will be reevaluated in left to right order
- SetValue should add the SharedVariable if a shared variable is passed in
- StartBehaviorTree should set the variable value instead of the reference
- Added word wrap within the log window
- Fixed conditional abort from triggering when the composite task is inactive under a parallel parent

- ErrorCheck should not check a BehaviorTree object
- Fixed Behavior.ExecutionStatus not being correct when the onBehaviorComplete event is triggered
- Fixed case where copying and pasting tasks wasn't copying the field value

---

#### **Version 1.5.7 Changes:**

- Added the Utility Selector task
- Behavior trees with an External Tree can be edited without having to select the External Tree within the project window
- Custom variables can be used by the reflection tasks
- Added an error check for project level variables referencing scene level objects
- GetPosition does not need allowEmptyTarget
- Has Received Event will only respond to events when the task is active or reevaluating
- Behavior Tree Inspector foldout will remember if it was folded out
- TaskCopier will create the shared SharedVariable if it doesn't exist
- Added the preference to allow breakpoints to select the tree that caused the breakpoint
- Use NETFX\_CORE instead of Metro || RT defines to allow IL2CPP be used with source on Windows Store
- Removed the CROSS\_PLATFORM\_INPUT definition within the input tasks
- Remember the execution status of a completed tree
- Fixed variable override from not correctly overriding the child variable in some cases
- Fixed infinite loop when disabling a paused tree
- Fixed Variable Mapping not pointing to the correct prefab object
- Fixed conditional abort from reevaluating when the grandparent task is a parallel task
- Removed support for Unity 4.6

---

#### **Version 1.5.6 Changes:**

- Added compatibility with Unity 5.4
- Deserialization optimizations
- Added NavMeshAgent tasks
- UnknownTask will retain the previous task data for easy replacement
- Moved task disabled state from editor to runtime (note: this will require you to disable previously-disabled tasks)
- TaskUtility startup optimizations
- ObjectPool optimizations
- Added duration and depthTest to Debug.DrawLine task
- Added modulo to FloatOperator and IntOperator tasks

- A destroyed behavior tree object will call OnBehaviorComplete
  - SetSharedGameObject can set the value to null
  - Added the find task API to the External Tree class
  - Fixed list field from being copied by reference when adding a new element
  - Fixed variable mappings from not working correctly when the variable overrides an external tree variable
- 

#### **Version 1.5.5 Changes:**

- Added a 5.3 build target
  - Editor gizmos work across 5.3 scenes
  - Editor GizmoManager optimizations
  - SetVariable will no longer check the type
  - Fixed overridden field from causing an infinite loop
  - Removed support for Unity 4.3
- 

#### **Version 1.5.4.1 Changes:**

- Fixed the behavior tree from not being reloaded correctly in the editor after stopping the game in Unity 5.3
- 

#### **Version 1.5.4 Changes:**

- Removed support for Unity 4.3
- Ensured compatibility with Unity 5.3
- Added per-GameObject variable mappings
- External trees referenced with the ExternalBehavior property will automatically add their local variables
- Double clicking on a reference task will open the tree by default
- JSON deserialization performance improvements
- Reflection tasks will use the current GameObject if there is no target set
- Fixed variables from not being overridden when within an array and being referenced from an external tree
- Fixed external tree variables from not being deserialized when the variables foldout is opened without the Behavior Designer editor opened
- Fixed the Gizmo View Type preference from not persisting
- Removed the deprecated InheritedFieldAttribute
- The following integration tasks or sample projects have been updated for the following assets:

- Third Person Controller

---

#### Version 1.5.3 Changes:

- Added variable mappings
- Added the variable list to the behavior tree and external tree inspector
- Added the option to specify individual variables on the Behavior Reference task
- Added a preference for the gizmos to only be drawn when the task is running
- Added OnBehaviorRestart task callback
- Added the SkipErrorCheck attribute – allows the task to be skipped from error checking
- Added the TaskName attribute – allows the task name to be changed to something other than the type name
- Fixed iOS 'Default Constructor Not Found' exception with runtime dll
- Fixed lower priority conditional abort from reevaluating self conditional tasks
- Fixed a paused behavior tree from not being able to be disabled
- Fixed the ClientRPC from sending a Transform
- Allow a pan on the graph if the command key is pressed and the MouseWheelScrolls preference is enabled
- SendEvent can specify individual behavior trees to send the event to
- Fixed SharedVariables from not being set properly when within a custom subclass of an external tree
- Deprecated InheritedFieldAttribute – use SharedVariables instead
- Added integration tasks and samples for the following assets:
  - Chronos
- The following integration tasks or sample projects have been updated for the following assets:
  - Core GameKit
  - DOTween
  - Playmaker
  - RFPS
  - Third Person Controller
  - uFrame

---

#### Version 1.5.2 Changes:

- Added Unity 5 networking support
- Added LayerMask tasks
- Added option to use CrossPlatformInputManager for the Input tasks
- Editor performance improvement
- Fixed not being able to deserialize enum lists with JSON serialization

- Fixed SharedVariables from not always being overridden with External Trees nested multiple levels
  - Fixed collapsed task connections from being able to be selected
  - The following integration tasks or sample projects have been updated for the following assets:
    - o Third Person Controller
- 

#### **Version 1.5.1 Changes:**

- Added the ability to reset values on tree restart
  - Added a toolbar popover which allows you to quickly move between referenced trees
  - Added the ability to specify a file name when creating a new Behavior Designer asset from the Create menu
  - Added a reference to the task within object drawers
  - Added compatibility with Unity 5.1
  - Fixed pink grid lines from being drawn with the PS4 platform selected
  - Fixed delete from not undoing properly in Unity 5
  - Fixed the local variables from not overriding the external variables when an external tree is assigned and the tree does not start when enabled while the editor is opened
  - Fixed invalid parameter error on SendMessage task which uses a Unity Object as its value
  - Removed deprecated field within binary deserialization
  - Added integration tasks and samples for the following assets:
    - o Blox 2
    - o Inventory Pro
  - The following integration tasks or sample projects have been updated for the following assets:
    - o Core GameKit
    - o Master Audio
    - o Playmaker
    - o RealisticFPS
    - o Third Person Controller
    - o Trigger Event Pro
    - o uScript
- 

#### **Version 1.5.0.1 Changes:**

- SendEvent can optionally send up to three arguments. Updated the SendEvent/HasReceivedEvent tasks
- Added a preference which allows you to set if the fields foldout by default
- Improved copy/paste - the disabled and collapsed status will be persisted
- Error check will ignore disabled tasks

- Correctly serializes a custom class derived from a List
- Replacing a composite task will keep all of the children
- Fixed pooling related error when restarting a behavior tree while it is in the middle of execution
- Fixed the tree from deserializing when it has already been deserialized and has an external tree attached
- Fixed the behavior tree object from being pooled when paused
- Fixed global variables from not deserializing correctly when using binary serialization
- Added integration tasks and samples for the following assets:
  - o Love/Hate
- The following integration tasks have been updated for the following assets:
  - o Core GameKit
  - o ICode
  - o Master Audio
  - o Realistic FPS Prefab

---

#### **Version 1.5 Changes:**

- Added the ability to execute an event on the behavior tree – can use Behavior.SendEvent or the Send Event task. Has Received Event will return success when the event was received
- Local behavior tree variables will automatically override the external tree variables
- Local behavior tree variables will not be in view only mode when an external tree is attached to the behavior tree component
- The task triggers a conditional abort or interruption is highlighted
- Behavior tree startup optimizations
- Added support for AnimatorCurve and Int16
- Added the task callbacks OnLateUpdate, OnFixedUpdate, and OnControllerColliderHit
- View Only Mode will not disable the right click Edit/Locate script options
- Event callback functions will not be called if the non-virtual function doesn't exist
- Added IBehavior.SetVariableValue to directly set the value of a SharedVariable
- Added the callback Behavior.OnBehaviorRestart
- Removed the callback Behavior.OnBehaviorEnd when the behavior tree is being restarted
- Optimizations for the basic tasks to prevent GetComponent from being called
- Force JSON serialization if the prefab instances preference is enabled – this prevents the Unity editor from lagging when comparing a large byte array across prefabs
- Behavior Reference task can be collapsed and it will carry over to the external tree
- HideInInspector fields no longer show in the task inspector
- Added integration tasks and samples for the following assets:
  - o Adventure Creator
- The following integration tasks have been updated for the following assets:
  - o Motion Controller
  - o PlayMaker

- Third Person Controller
- Vectrosity

---

#### **Version 1.4.7 Changes:**

- Added a separate Unity 5 Asset Store package
- Set the minimum Unity requirement to 4.3
- Relocated the integration task downloads to the samples page
- Added a set of light tasks
- A value can be used within SendMessage
- Removed deprecated task fields
- Added GetMouse tasks
- Fixed null reference exception when deleting a base SharedVariable reference from another task
- History arrow buttons will select the UnityEngine.Object
- The following integration tasks have been updated for the following assets:
  - Motion Controller
  - Third Person Controller

---

#### **Version 1.4.6.1 Changes:**

- Workaround for IL2CPP not implementing MakeGenericType yet. This workaround reduces the usage of MakeGenericType and will allow 64-bit iOS builds to work in all situations. When Unity adds MakeGenericType support to IL2CPP this workaround can be removed (though it doesn't harm anything)

---

#### **Version 1.4.6 Changes:**

- Tasks within external assemblies will now be found
- Added a preference to always show the gizmos, only show the gizmos when selected, or never
- Wait task can optionally choose between a random min and max wait time
- Collision and trigger tasks can optionally compare against a tag
- Added a "Locate Script" option to the task inspector and task right click menu
- Added Physics/Physics2D tasks
- Added uint serialization
- Added an integrations link to the welcome screen
- Use TryGetValue instead of ContainsKey and the accessor for better performance
- Fixed NullReferenceException when using a LowerPriority conditional abort and a parent decorator resumes the branch without first moving to a new branch

- Added integration tasks and samples for the following assets:
  - o DOTween
  - o LeanTween
  - o Third Person Controller
  - o uFrame
- Updated integration tasks for the following assets:
  - o Realistic FPS Prefab
  - o UFPS

**This will be the last release that supports Unity 4.0, 4.1, and 4.2.**

---

#### **Version 1.4.5.1 Changes:**

- Fixed an error when unpausing a paused tree
  - Reflection tasks will search in non-Unity DLLs
  - Reflection tasks can use a string as a parameter
  - Added Rigidbody2D position and rotation getters/setters
  - Updated integration tasks for the following assets:
    - o Realistic FPS Prefab
    - o UFPS
- 

#### **Version 1.4.5 Changes:**

- Added the ability to select global variables to the Variable Synchronizer
- Added the ability to select the Task Execution Type from the Behavior Manager – allows you to run a maximum number of tasks every tick instead of stopping when a duplicate task is encountered.
- Added a Selector Evaluator task – use to reevaluate every child every tick
- Task.OnDrawGizmos only gets called if the behavior tree is active
- Tasks now support the byte field
- Added OnCollision/OnTrigger conditional tasks
- Runtime and physics callback optimizations
- Fields of type SharedVariable can now be inherited
- Place the tree in View Only mode when editing a prefab instance – can be disabled in the preferences
- Fixed nested Parallel Selectors from creating orphaned children during a conditional abort
- Fixed the GUI textures from being dark with the linear color space set
- Fixed a null reference sometimes caused when deleting a SharedVariable from the variable inspector
- Fixed SetVariable not persisting the new variable change to the tasks



- Copy and pasting reflection tasks no longer shallow copies the SharedVariables
- Editor tree loading cleanup
- Can change the Unity layers and the LayerMask will accurately reflect that change without having to restart
- Added tasks to synchronize individual PlayMaker and Behavior Designer variables
- Updated integration tasks for the following assets:
  - o PlayMaker
  - o Simple Waypoint System
- Added integration tasks and samples for the following assets:
  - o Anti-Cheat Toolkit
  - o Cinema Director
  - o Pool Boss
  - o Realistic FPS Prefab
  - o Uni2D

---

#### **Version 1.4.4 Changes:**

- Added a SharedGenericVariable to allow you to select the variable type within the inspector
- Added an update checker
- Added the ability to create a new SharedVariable from the create menu
- Added indentation to JSON serialization
- Added an option to the Start Behavior Tree task which allows you to wait for the behavior tree to complete
- Added the following callbacks to the Behavior component:
  - o OnBehaviorStart
  - o OnBehaviorEnd
- Tasks can include a private constructor
- Reflection tasks look in more assemblies than just the C# assembly
- Correctly position tasks included in external trees multiple levels deep
- Disable the tree on startup if the root task is disabled
- Task list sorting improvements
- Improved error checking with SharedVariables
- Improved the format of the task inspector when showing fields with SharedVariables multiple levels deep
- Removed ShowExternalTree preference
- Removed deprecated OnSceneGUI code
- Added basic tasks for the following classes:
  - o Debug
  - o ParticleSystem
  - o PlayerPrefs
  - o String

- Added integration tasks and samples for the following assets:
    - o Particle Playground
    - o SECTR
    - o Vectrosity
- 

#### **Version 1.4.3 Changes:**

- Switched the base class of Task and SharedVariable to object from ScriptableObject
  - Added support for structs
  - Removed deprecated serialization code
  - Fixed Visual Studio from not correctly resolving the runtime with Unity 5
  - Fixed conditional aborts not reevaluating a conditional task when there are multiple sibling conditional tasks and the first conditional task triggers an abort
  - Check for serialization at a higher level when modifying shared variables so the external tree can be deserialized
  - Fixed binary and JSON serialization so they can correctly deserialize global variables in all cases
  - Added more basic tasks for the following classes:
    - o Input
    - o Quaternion
    - o Transform
    - o Vector2
    - o Vector3
  - Added integration tasks and samples for the following assets:
    - o Control Freak
    - o Curvy
    - o Glow Effect
    - o Trigger Event Pro
    - o UniStorm
- 

#### **Version 1.4.2.1 Changes:**

- Added object pooling to the Behavior Manager
- Added behavior.BehaviorName and behavior.BehaviorDescription properties
- Task.OnEnd and Task.OnBehaviorComplete will be called from OnApplicationQuit
- Remove all outstanding conditional aborts before the tree is disabled
- Fixed JSON exception when removing a value from an enum with that value selected
- Fixed exception when inheriting Shared Variables into multi-nested external trees
- The following integration tasks have been updated:
  - o NGUI

- PlayMaker
- Simple Waypoint System
- Ultimate FPS (UFPS)

---

#### **Version 1.4.2 Changes:**

- Added a variable synchronizer component
- Added a button to take screenshots of your tree
- Added IntSlider and FloatSlider object drawers
- Object drawer attributes can be used on SharedVariables
- Draw the LayerMask field with MaskField instead of LayerField
- RotateTowards has the option of only rotating on the Y axis
- Improved the location of the error icon within the task inspector
- Startup performance improvements - Don't deserialize external trees if they have already been deserialized
- Fixed a null reference when a task is being interrupted and is disabled
- Variable name mapping wasn't being updated correctly when a variable name changed
- Changed OnAwake to OnStart for basic tasks so they can use SharedVariables
- Set the owner of the task immediately during deserialization
- New integration task and sample project:
  - Camera Path Animator
- The following integration tasks have been updated:
  - AI For Mecanim
  - Core GameKit
  - PlayMaker

---

#### **Version 1.4.1 Changes:**

- Recursive conditional aborts improvements
- Startup performance improvements – don't initialize behavior trees that have already been initialized
- OnBehaviorComplete will be triggered on all tasks when the Behavior Manager is destroyed
- Fixed the global variables asset file from not being able to be created
- Can select tasks while in a prefab
- Fixed Conditional evaluator task from throwing null reference exceptions
- Prevented the variable selection from persisting across newly opened trees
- The log task respects the logError flag
- Tasks can now run public coroutines versus having to be private
- The following integration tasks and the corresponding sample project have been updated:

- 2D Toolkit
- AI For Mecanim
- Core GameKit
- Dialogue System
- Final IK
- Master Audio
- Motion Controller
- NGUI
- PoolManager
- Simple Waypoint System
- UFPS
- uSequencer

---

### Version 1.4 Changes:

- Runtime source code is now located on the [runtime webpage](#).
- Added a realtime error checker
- New Variable inspector – can rename, change the type, and rearrange the Shared Variables
- Added Object Drawers – similar to Property Drawers
- Added an option to automatically inherit SharedVariables into an external tree
- Conditional evaluator task now has the option of being reevaluated
- Added recursive conditional aborts
- Added the implicit operator to Shared Variables
- Added Task colors
- Added the Behavior Game GUI to view the currently active tasks
- Added BehaviorTree.GetActiveTasks
- Added the RequiredComponent task attribute
- Added the RequiredField task field attribute
- Added the task type to the task inspector if the name is different
- Trigger/Collision events will now be called on active conditional abort tasks and active conditional evaluators
- Deprecated assignment of Unity components within the tasks to ensure compatibility with Unity 5
- Deprecated Task.OnBehaviorReset, use Task.OnBehaviorComplete
- Deprecated Task.OnSceneGUI, use Task.OnDrawGizmos
- Updated all of the tasks to use the new features
- In a future release, the SharedVariable and Task classes will no longer be derived from ScriptableObject and will instead be derived from System.object.
- The following integration tasks have been updated:
  - PlayMaker
  - uScript

- The following sample projects have been updated:
    - o CTF
    - o RTS
    - o MiniGauntlet
    - o PlayMaker
    - o uScript
- 

#### **Version 1.3.6 Changes:**

- Integration tasks now located on the [integrations webpage](#)
  - 2D Toolkit integration
  - Simple Waypoint System integration
  - Selected tasks are correctly remembered when changing focus between windows
  - BehaviorManager will respect changes made to the update interval after it has started
  - The following integration tasks have been updated:
    - o Core GameKit
    - o Master Audio
    - o PlayMaker
    - o uScript
- 

#### **Version 1.3.5 Changes:**

- New option to set the type of serialization: binary or JSON. This option can be toggled from the preferences
- AI For Mecanim integration
- The following BehaviorTree fields will change protection levels from public to private. The properties should be used instead:
  - o behaviorTree.externalBehavior (use behaviorTree.ExternalBehavior)
  - o behaviorTree.group (use behaviorTree.Group)
  - o behaviorTree.startWhenEnabled (use behaviorTree.StartWhenEnabled)
  - o behaviorTree.pauseWhenDisabled (use behaviorTree.PauseWhenDisabled)
  - o behaviorTree.restartWhenComplete (use BehaviorTree.RestartWhenComplete)
  - o behaviorTree.logTaskChanges (use behaviorTree.LogTaskChanges)
- Editor optimizations
- More startup optimizations
- Simplified the code that determines when a tree should be loaded in the editor
- A disabled task is now actually disabled instead of just returning success
- Restart Behavior Tree task can be used to restart the current behavior tree without throwing any errors

- Added the task name to the log when LogTaskChanges is enabled
  - Added OnDrawGizmos virtual method to the tasks
- 

#### **Version 1.3.4a Changes:**

- Updated the Core GameKit tasks by replacing AttackOrHitPointsAdd with AttakOrHitPointsChange – brings these tasks up to date with the latest Core GameKit release (3.2.2.2)
  - Updated the deprecated method calls in the Motion Controller – brings these tasks up to date with the latest Motion Controller release (1.24)
  - Bugfixes
- 

#### **Version 1.3.4 Changes:**

- Serialize the data in binary instead of JSON – deprecated JSON serialization
  - Added Master Audio tasks
  - Added NGUI tasks
  - Added math related tasks
  - External trees will show in the behavior tree dropdown box once loaded
  - Copying/pasting tasks will stay relative to the window position
  - Fixed error appearing in certain circumstances when loading external behavior trees within each other
- 

#### **Version 1.3.3 Changes**

- Relocated Behavior Designer to Tools/ instead of Window/
- Added reflection tasks:
  - o Invoke Method
  - o Set Field Value
  - o Set Property Value
  - o Get Field Value
  - o Get Property Value
  - o Compare Field Value
  - o Compare Property Value
- Added Conditional Evaluator decorator
- Added Final IK integration tasks
- Added Pool Manager integration tasks
- Added the following basic tasks:

- GameObject - Destroy
- GameObject - DestroyImmediate
- GameObject - Find
- GameObject - FindWithTag
- SharedGameObjectToTransform
- SharedTransformToGameObject
- Added the ability to Play, Pause, Stop, and Step within the Behavior Designer window
- Added an option to replace the task by right clicking
- Added an icon on the task when a conditional abort is selected
- Added conditional abort execution status indicators
- Always copy the tasks if they came from a prefab
- Locking the tree used to cause a null reference exception when switching selections in certain circumstances

---

#### **Version 1.3.2 Changes:**

- Added Motion Controller integration – the GoblinLife sample project provides an excellent example of a complete behavior tree
- Added Ultimate FPS integration
- Added uSequencer integration
- Added Core GameKit integration
- Unknown tasks will appear red
- Fixed Windows Store build errors caused by version 1.3.
- Composite tasks with a lower priority conditional abort can have a decorator above it
- Added a signal timeout to the uScript task – used if the Behavior Tree Signal component isn't added immediately
- Added the PlayMaker Run Conditional FSM task for use by conditional aborts
- Added the uScript Run Conditional Graph task for use by conditional aborts
- Performance improvements when stopping a third party task
- General editor performance improvements

---

#### **Version 1.3.1 Changes:**

- Added Behavior.FindTask and Behavior.FindTaskWithName
- Added Assets/Create/Behavior Designer context menu – can create external behavior trees, actions, and conditional tasks
- Added Component/Behavior Designer context menu
- Added HelpURL links for PlayMaker SendEvent/StopFSM tasks
- Start/Stop Behavior Tree tasks now use the correct GameObject if behaviorGameObject is null

- Correctly detect null elements in an array when serializing
- Perform a deep copy if the root task instance already exists. This will happen with spawning multiple prefabs
- Deserialize the external behavior tree when starting up if necessary

---

### Version 1.3 Changes:

- Conditional Aborts
- Global Variables
- Custom Shared Variables
- Added a grid background
- Added a compact mode
- Added a field for the External Behavior Tree on the Behavior Tree component
- Added history buttons on the top of the window
- Added a right click menu option to edit the script
- Added "Open Tree Reference On Task Double Click" preference
- Added "Mouse Wheel Scrolls" preference
- Autoscrolls the graph when dragging an element and the mouse reaches the graph edge
- Performance improvements when enabling a behavior tree
- Added SharedObjectList, SharedGameObjectList, SharedTransformList variable
- Added a set of Compare Shared Variable conditional tasks
- PlayMaker integration improvements:
  - o StartFSM can now synchronize variables
  - o StartFSM has the option of returning success immediately
  - o Added Stop FSM and Send Event tasks
  - o Added Start and Stop Behavior Tree actions
- uScript integration improvements:
  - o StartGraph has the option of returning success immediately
  - o Added Start and Stop Behavior Tree nodes
- Serializes generic lists correctly (both in tasks and shared variables)
- Improved error reporting when a behavior tree starts
- Windows Store and Windows Phone can now build with .Net Core enabled
- Added behaviorTree.ExecutionStatus
- Removed the deprecated tasks Start\_PlayMakerFSM and Start\_uScript
- Deprecated many function names that begin with a lowercase. You'll receive a warning if you are using the function

---

### Version 1.2.7 Changes:



- Fixed a bug that prevented you from adding new variables in Unity 4.5
- Inherited fields weren't being removed when the external tree is no longer active
- Vector.ToString was removing precision during serialization
- Can now run multiple uScript/PlayMaker tasks under a single parallel task
- Fixed AOT iOS exception when deserializing an Enum array
- Added a grouping option to the Start/Stop Behavior Tree task
- End On Failure wasn't working properly if Repeat Forever was selected on the Repeater task

---

#### **Version 1.2.6 Changes:**

- Added a repeat forever option to the Repeater task
- Added a new basic task: SendMessage
- Correctly serializing the breakpoint, disabled, and collapsed state of the task
- New tasks weren't always being ordered correctly relative to other children
- SharedVariables were being reset when the game started under certain conditions
- Worked around a Unity bug that caused LayerMask fields to be set incorrectly
- The lock no longer causes exceptions when no behavior tree is selected
- Referencing a single task no longer causes runtime exceptions
- Added a check within RandomSelector/RandomSequence to prevent popping from an empty stack

#### **Required Changes:**

- Task.OnChildRunning has been deprecated. Use OnChildStarted
- BehaviorReference.getBehaviorTrees has been deprecated. Use GetBehaviorTrees

---

#### **Version 1.2.5 Changes:**

- Updated the welcome screen link to the forums – now located at <http://www.opsive.com/forum>
- Added Boo support
- Added 2D OnCollision/Trigger events
- Added BehaviorManager.Tick(BehaviorTree) to allow for manually ticking a specific behavior tree
- Fixed issues with SetVariable
- The editor scrollbars will snap back to the center when a behavior tree has been loaded
- Use a node offset instead of position – should help with the performance for some users
- Serialize private fields on a base class marked with SerializeField
- Improves handing of tasks that already exist which have a class name that has been renamed to be different from the file name

- Fixed leaking objects editor warnings
- Better check for null on deserialization
- Can compile for WP8/Windows Store using the DLL – no longer required to use the source code
- Updated Unity defines
- Improves editor performance when loading behavior trees
- Lots of Serialization/deserialization fixes

**Note:**

- Because of the change to use the node offset instead of the position inside graphs, if an external behavior tree is loaded during runtime by an behavior tree reference task then the tasks may overlap each other. To fix this just open up the external behavior tree and make a minor change (such as dragging a node around) to force it to save.

---

**Version 1.2.4 Changes:**

- Prefabs now work correctly: instantiating, applying, and reverting will stay consistent with the Unity standard
- Handles renamed/removed tasks better – will show an unknown task and preserve the children
- Moved the composite and decorator categories to the top of the task list
- Remembers which categories have been expanded after closing the window
- Watched variables with a null value will no longer throw exceptions
- Improved the handing of null values when they are serialized

---

**Version 1.2.3 Changes:**

- Added over 215 basic tasks
- Added a preference to open the task inspector on a task double click
- Fields starting with “m\_” will no longer show “M “ within the task inspector
- Fixed linked tasks not synchronizing properly when referenced/dereferenced
- Removed deprecated shared/synchronized attributes
- Improved serialization: custom classes are now serialized
- The editor now looks for tasks within the Assembly-UnityScript and Assembly-UnityScript-firstpass assemblies

---

**Version 1.2.2 Changes:**

- Placed the runtime source code in the Unity package located at /Behavior Designer/RuntimeSource.unitypackage. This was done to keep the Behavior Designer package neater.
- Added Dialogue System for Unity integration
- Added the Tooltip attribute –place on task fields to see a tooltip within the task inspector
- Added the Restart behavior Tree Task
- No longer serializing behavior id – this id was causing the prefab/asset file to be changed even though there were no real changes – improves version control
- Minor bugfixes

#### **Required Changes:**

- Deprecated the task Start\_PlayMakerFSM. Use StartFSM.
- Deprecated the task Start\_uScript. Use StartGraph.
- Deprecated BehaviorTree.enableBehavior. Use BehaviorReference.EnableBehavior
- Deprecated BehaviorTree.disableBehavior. Use BehaviorTree.DisableBehavior
- Deprecated BehaviorManager.tick. Use BehaviorManager.Tick

---

#### **Version 1.2.1 Changes:**

- No “real” new features or fixes, corrected the meta files so the scripts associate correctly when importing the sample projects or Movement Pack.

---

#### **Version 1.2 Changes:**

- External behavior trees will load in the editor when running
- Multiple behavior trees can be referenced through the behavior tree reference task
- Can reference an external behavior tree through script using behaviorTree.externalBehavior
- Added search to the task list
- Added pause/resume signals to uScript
- Added pause/resume/end events to PlayMaker
- Added watched variables
- Can show the task description within the editor with the TaskDescription attribute (new pref added)
- Tasks can be disabled through the editor – hover over the task to show this option
- Parent tasks can be collapsed – hover over the task to show this option
- The size of the connection tab on the tasks has slightly been increased
- Increased the hit area of the task connection line to make clicking on it easier
- Variables from a referenced tree will be loaded into the parent tree
- Prevented shared variables from being shared by more than one behavior tree

- Serialization fixes

#### Required Changes:

- Deprecated BehaviorReference.externalBehavior. Use the array BehaviorReference.externalBehaviors instead
- Deprecated Start\_PlayMakerFSM.eventName – use Start\_PlayMakerFSM.startEventName instead

---

#### Version 1.1.2 Changes:

- Relocated the core runtime files to Plugins/Behavior Designer to get around Unity bug 599473.
- The active tasks will now receive OnCollision/OnTrigger callbacks
- The node connection line will show even at the furthest zoom level
- Private task fields will show in the inspector if they have the SerializeField attribute
- Builds correctly on WINRT
- Parent behavior trees will consume the external behavior tree's shared variables
- Serialization fixes
- Changed the default values for the repeater task
- Added a preference to turn off the task fading
- Minor bugfixes

#### Required Changes:

- **The location of the core runtime files has changed.** Before you update remove the entire Behavior Designer/runtime folder.

---

#### Version 1.1.1 Changes:

- Includes the runtime source code!
- Improved UnityEngine.Object serialization
- Improved the running visualization: tasks will smoothly fade after they get done running
- Bugfixes

---

#### Version 1.1 Changes:

- Added variables
- The “Save” button now exports to a .asset file
- Can specify the time that the Behavior Manager updates
- Tasks can be categorized in the editor with TaskCategory

- Added InheritedField attribute
- Added an instant property to tasks
- Bugfixes

#### **Required Changes:**

- **The data format for the behavior trees has changed.** Run the Behavior Designer Update tool on each behavior tree to update to the most recent format.
  - Deprecated the SharedField and SyncedField attributes. The new variables feature replaces these in functionality
  - Deprecated the task External Behavior Tree – use the task Behavior Tree Reference instead
- 

#### **Version 1.0.3 Changes:**

- Serialization fixes
  - Renamed “link” to “reference” when referencing other tasks within the graph
- 

#### **Version 1.0.2 Changes:**

- Added Behavior Designer gizmos (can be disabled within the preferences)
  - Proper coroutine support within tasks
  - Serialization fixes
  - Sort the tasks alphabetically, ignoring namespaces
  - Support the “Delete” keyboard command
  - Added the time to logging
- 

#### **Version 1.0.1 Changes:**

- In Unity 4.3 Resources.LoadAssetAtPath causes unnecessary warnings/errors if a project was upgraded from a previous Unity version. This fix removes that call.