

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 1.3.5 New Features/Fixes:

- 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 New Features/Fixes:

- 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 New Features/Fixes:

- 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 New Features/Fixes:

- 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:
 - o GameObject - Destroy
 - o GameObject - DestroyImmediate
 - o GameObject - Find
 - o GameObject - FindWithTag
 - o SharedGameObjectToTransform
 - o 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 New Features/Fixes:

- 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 New Features/Fixes:

- 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 New Features/Fixes:

- 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 Fixes:

- 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 Fixes:

- 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 Fixes:

- 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 Fixes:

- 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 handling of null values when they are serialized

Version 1.2.3 Features/Fixes:

- 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 Fixes:

- 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 Fixes:

- 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 Features / Fixes:

- 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 Features / Fixes:

- 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 Features / Fixes:

- 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 Features / Fixes:

- Added variables ([documentation](#))
- The “Save” button now exports to a .asset file ([documentation](#))
- Can specify the time that the Behavior Manager updates ([documentation](#))
- Tasks can be categorized in the editor with TaskCategory ([documentation](#))
- Added InheritedField attribute ([documentation](#))
- Added an instant property to tasks ([documentation](#))
- 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 Features / Fixes:

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

Version 1.0.2 Features / Fixes:

- 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 Features / Fixes:

- 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.