Unity Composition Overview

PiRho Soft

Introduction

Welcome to the documentation website for the PiRho Soft Composition Framework! On this site you will find a comprehensive set of materials for learning and becoming familiar with the framework. We have several tutorials that cover all the basics, in depth topics on the main systems, a manual containing pages for each object type, and a complete api reference. Use the navigation menu on the left to find what you are looking for, or use the search box to filter the menu to only show pages containing a certain term.

At the top right of any page you can find a Pdf download link. Next to that is a link to our Discord server where we will be available to quickly respond to any questions or suggestions (whether you have purchased the asset yet or not).

We have a Roadmap in the navigation menu as well that we will be keeping up to date as our development plans evolve. This includes a bugs section that contains a list of the currently known issues. Below that is the Log Descriptions section. This is a complete glossary for all the messages the framework can print to the console with more information about what they mean and how to deal with them.

For beginners, your best bet is to start with the Getting Started Tutorial. This will walk you through creating a basic game using graphs, variables, and bindings. After that, there are more tutorial videos that can be viewed in any order depending on what you are looking to accomplish. After the tutorials, the various topics provide more detailed information about specific systems, such as tips for debugging graphs, how to implement your own nodes, and how to define objects in the editor.

In the manual you will find a page for each of the component and asset types. These pages can be accessed from inside the Unity editor using the  icon in the inspector. The manual explains how to use each object including the list of properties it supports. The reference pages expand on the manual with complete coverage of the code api for every type.

If you are struggling with anything, don't be afraid to ask for help. And if you make something cool, we would love to see it in the gallery!

Roadmap

The following is a list of planned features for future releases. This will be periodically updated with more features, additions, and fixes and more specific information (like release dates) as our plans develop. If there is something you would like to see that isn't listed here, or something that is listed that you would like to voice your support for (and potentially increase its priority), let us know!

Features

- First class integration with DOTS
- Update custom editors to support UIElements
- Support for the new InputSystem
- Intellisense/auto complete in VariableReference and Expression editing
- Broader support for exposing, editing, and animating properties using graphs
- Precompilation of entire graphs and objects for enhanced performance (competitive with hand written code)
- Graph analysis and display for inspecting how variables and bindings are being used

Additions

- More logging and debugging info, including a visualizer for bindings and expressions
- Live display of graph execution information (iterations, frame count, elapsed time)
- Various visual enhancements of the graph window, including better connection drawing and animated view changes
- More built in node and binding types
- More transition effects
- Support for number formatting and expression evaluation in messages
- `Other` type for VariableValue so non Object derived types can be stored using a ClassMap
- Reflection fallback for `Object/Other` VariableValues that aren't IVariableStores and don't have a ClassMap
- DictionaryAdapter and ObjectAdapter (using ClassMaps) for use with MappedVariable or IVariableStore implementations
- More example projects, including ones using these types that don't have an example: TimeScaleNode, TextInputBinding, SetAnimationParameterNode, PlayAnimationStateNode, PlayTimelineNode, ImageColorBinding, CutoffTransition Distort, PixelateTransition, CreateScriptableObjectNode, and ResetVariableNode
- MenuBinding that works like ListBinding but uses MenuItemTemplates
- Potentially add commands or operators for testing enum flags in expressions
- Break on error option in graph execution
- Simplified and documented interface for extending expressions with custom keywords and

operators

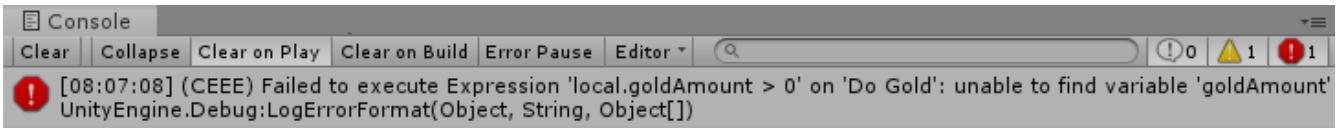
- Store VariableValue enums as int so they don't box (i.e allocate)
- Expose access restrictions for set, add, and remove to MappedVariable for lists
- Context sensitive mouse cursors in graph window (disabled due to odd AddCursorRect behaviour with zoom)
- Improve custom binding interface with a generic base class that looks up the sibling component the binding is acting on

Bug Fixes

- the outputs in the connection lists for MockupGraph and MockupNode must have unique names but this is not enforced by the editor
- pasting a node in a graph, then undoing, then redoing will not maintain connections
- right clicking in the graph window then right clicking again in a different place sometimes pans the view
- when zoomed tooltips in the graph window don't show up in the correct place
- the VariablePool inspector doesn't repaint when changing the constraint from a SelectionControl until mouse over
- lists in a VariableStore are not assigned a definition so statically typed lists (i.e ListAdapters) cannot have items added
- auto scrolling in MenuInput doesn't take into account different anchors
- switching focused menus with MenuInput is based on item index without accounting for scrolling - it should be based on the item position instead
- key events are not repeated when holding down a key
- GetVariableNames in BindingRoot does not include names from _parent

Log Descriptions

The following lists all the warning and error messages that may be printed to the Console by the Composition Framework. Each message has an associated id that corresponds to an entry in this list for easy searchability. For example the following image shows an error with id CEEE. Searching this page for an id will take you to the entry which will explain in more detail what the message means, the effect on execution, and potential ways to fix it.



General Tips

When dealing with VariableReference errors, the Watch Window can be extremely useful. Global variables can be investigated at any time and variables on a specific InstructionGraph can be investigated by setting a breakpoint on a node.

Expression errors can be harder to track down. If the text box containing the Expression is red, the issue is with the actual structure of the Expression and details will be displayed. In this case the error state will update as the Expression is modified so it is easy to see when the issue has been resolved. If the Expression is valid and an error is printed at runtime, the issue is usually related to a VariableReference and the above tip for using the Watch Window applies.

Warnings

CAANPIL

Unable to wait on Animation Player '{0}': the Animation Clip '{1}' was set to loop and would have never finished

{0}	The GameObject containing the AnimationPlayer that triggered the warning.
{1}	The AnimationClip that is being played.

Effect

The AnimationClip will still play successfully but the wait setting will be ignored.

Resolution

Either tell the AnimationPlayer to play the animation without looping (i.e call *PlayAnimation* instead of *PlayAnimationAndWait* or clear the *WaitForCompletion* flag if using a PlayAnimationNode) or change the AnimationClip to one that doesn't loop.

CAAUPIL

Unable to wait on Audio Player '{0}': the Audio Clip '{1}' was set to loop and would have never finished

{0}	The GameObject containing the AudioPlayer that triggered the warning.

{1}	The AudioClip that is being played.

Effect

The AudioClip will still play successfully but the wait setting will be ignored.

Resolution

Either tell the AudioPlayer to play the sound without looping (call *PlaySound* instead of *PlaySoundAndWait* or clear the *WaitForCompletion* flag if using a PlaySoundNode) or change the AudioClip to one that doesn't loop.

CBBBWT

Bar Binding '{0}' has an Image with a type that is not 'Filled'

{0}	The GameObject containing the BarBinding that triggered the warning.

Effect

The Image will have its *type* set to `Filled` with default values for the other fill related properties.

Resolution

Change the `Image Type` of the Image to `Filled`.

CBLBMT

Unable to create item for List Binding '{0}': the Template is null

{0}	The ListBinding that triggered the warning.

Effect

The ListBinding will do nothing.

Resolution

Set the *Template* property of the ListBinding.

CDONIO

Unable to disable object for node '{0)': the object '{1}' is not a GameObject, Behaviour, or Renderer

{0}	The DisableObjectNode that triggered the warning.
{1}	The VariableReference assigned to *Target* that resolved to an invalid type.

Effect

The InstructionGraphNode will do nothing.

Resolution

Change *Target* so it resolves to either a GameObject, Behaviour, or Renderer.

CEBFC

Unable to bind enabled state for binding '{0}': the Command '{1}' failed with error '{2}'

{0}	The EnableBinding that triggered the warning.
{1}	The name of the command that failed.
{2}	The error message reported by the command.

Effect

The enabled state of *Object* will not be changed.

Resolution

If the reported error ({2}) indicates a parameter is invalid, the *Condition* Expression is calling the command incorrectly. Otherwise, the command itself has an error and needs to be edited directly. The message should provide details on how to go about fixing the issue.

CEBFE

Unable to bind enabled state for binding '{0}': the expression '{1}' failed with error '{2}'

{0}	The EnableBinding that triggered the warning.
{1}	The statement in the *Condition* Expression that failed.
{2}	The error message reported by the expression.

Effect

The enabled state of *Object* will not be changed.

Resolution

The reported error ({2}) should provide details on how to go about fixing the *Condition* Expression.

CEBIO

Unable to bind enabled state for binding '{0)': the object '{1}' is not a GameObject, Behaviour, or Renderer

{0}	The EnableBinding that triggered the warning.
{1}	The Object assigned to *Object*.

Effect

The enabled state of *Object* will not be changed.

Resolution

Change *Object* to either a GameObject, Behaviour, or Renderer.

CEBIV

Unable to bind enabled state for binding '{0}': the expression '{1}' did not evaluate to a bool

{0}	The EnableBinding that triggered the warning.
{1}	The final statement in the *Condition* Expression.

Effect

The enabled state of *Object* will not be changed.

Resolution

Change *Condition* so that the final statement results in a bool value, potentially by using a comparison operation.

CEBMV

Unable to bind enabled state for binding '{0}': the expression is empty

{0}	The EnableBinding that triggered the warning.

Effect

The enabled state of *Object* will not be changed.

Resolution

Set the *Condition* Expression.

CEIR

'{0}' expected the Expression '{1}' to return type '{2}' but it instead returned type '{3}'

{0}	The Object that executed the Expression.
{1}	The final statement of the Expression that was expected to result in a specific type.
{2}	The VariableType the caller expected from the result of the expression.
{3}	The actual VariableType of the result of the expression.

Effect

The caller ({0}) will still be returned the result and how that result is used is dependent on the specific scenario.

Resolution

Change the expression defined on the caller ({0}) so that the final statement results in the expected type ({3}).

CEONIO

Unable to enable object for node '{0)': the object '{1}' is not a GameObject, Behaviour, or Renderer

{0}	The EnableObjectNode that triggered the warning.
{1}	The VariableReference assigned to *Target* that resolved to an invalid type.

Effect

The EnableObjectNode will do nothing.

Resolution

Change *Target* so it resolves to either a GameObject, Behaviour, or Renderer.

CEXBFC

Unable to bind text for binding '{0}': the Command '{1}' failed with error '{2}'

{0}	The ExpressionBinding that triggered the warning.
{1}	The name of the command that failed.
{2}	The error message reported by the command.

Effect

The TMP_Text will be hidden and cleared.

Resolution

If the reported error ({2}) indicates a parameter is invalid, the *Expression* Expression is calling the command incorrectly. Otherwise, the command itself has an error and needs to be edited directly. The message should provide details on how to go about fixing the issue.

CEXBFE

Unable to bind text for binding '{0}': the expression '{1}' failed with error '{2}'

{0}	The ExpressionBinding that triggered the warning.
{1}	The statement in the *Expression* Expression that failed.
{2}	The error message reported by the expression.

Effect

The TMP_Text will be hidden and cleared.

Resolution

The reported error ({2}) should provide details on how to go about fixing the *Expression* Expression.

CEXBMV

Unable to bind text for binding '{0}': the expression is empty

{0}	The ExpressionBinding that triggered the warning.

Effect

The TMP_Text will be hidden and cleared.

Resolution

Set the *Expression* Expression.

CIGNIA

Failed to assign to variable '{0}' from node '{1}': the variable has an incompatible type

{0}	The VariableReference that resolved to an invalid VariableType.
{1}	The InstructionGraphNode that attempted to assign to the variable.

Effect

The assignment will not be performed.

Resolution

Change the VariableReference on the InstructionGraphNode so it either resolves to a variable with the correct type or resolves to a variable that can be assigned any type.

CIGNIE

Failed to resolve variable '{0}' on node '{1}': the variable has enum type '{2}' and should have enum type '{3}'

{0}	The VariableReference that resolved to an invalid enum type.
{1}	The InstructionGraphNode that attempted to resolve the variable.

Effect

The effect depends on the InstructionGraphNode. Usually it will skip performing its action but it may use a default value instead.

Resolution

Change the VariableReference on the InstructionGraphNode so it resolves to a variable with the correct enum type.

CIGNIO

Failed to resolve variable '{0}' on node '{1}': the object '{2}' is a '{3}' and cannot be converted to a '{4}'

{0}	The VariableReference that resolved to an invalid object type.
{1}	The InstructionGraphNode that attempted to resolve the variable.
{2}	The Object that was resolved but is not the correct type.
{3}	The type of the resolved Object.
{4}	The type the InstructionGraphNode expected.

Effect

The effect depends on the InstructionGraphNode. Usually it will skip performing its action but it may use a default value instead.

Resolution

Change the VariableReference on the InstructionGraphNode so it resolves to a variable with the correct object type.

CIGNIO

Failed to resolve variable '{0}' on node '{1}': the value is a '{2}' and cannot be converted to a '{3}'

{0}	The VariableReference that resolved to an invalid object type.
{1}	The InstructionGraphNode that attempted to resolve the variable.
{2}	The type of the resolved value.
{3}	The type the InstructionGraphNode expected.

Effect

The effect depends on the InstructionGraphNode. Usually it will skip performing its action but it may use a default value instead.

Resolution

Change the VariableReference on the InstructionGraphNode so it resolves to a variable with the correct type.

CIGNIV

Failed to resolve variable '{0}' on node '{1}': the variable has type '{2}' and should have type '{3}'

{0}	

Effect

Resolution

CIGNMA

Failed to assign to variable '{0}' from node '{1}': the variable could not be found

{0}	

Effect

Resolution

CIGNMV

Failed to resolve variable '{0}' on node '{1}': the variable could not be found

{0}	

Effect

Resolution

CIGNROA

Failed to assign to variable '{0}' from node '{1}': the variable is read only

{0}	

Effect

Resolution

CIMMV

Unable to get text from Message '{0}': the Variable '{1}' could not be found

{0}	

Effect

Resolution

CNBIV

Failed to resolve variable '{0}' on binding '{1}': the variable has type {2} and should have type 'Int' or 'Float'

{0}	

Effect

Resolution

CNSLS

Unable to load scene for {0}: the scene '{1}' could not be found. Make sure this variable refers to an int or a string

{0}	

Effect

Resolution

CNSUS

Unable to unload scene for {0}: the scene '{1}' could not be found. Make sure this variable refers to an int or a string

{0}	

Effect

Resolution

CSCIE

Failed to expand item {0}: the variable '{1}' is not an IVariableList

{0}	

Effect

Resolution

CTMIA

this TransitionRenderer has already been added

{0}	

Effect

Resolution

CTMIR

this TransitionRenderer has not been added

{0}	

Effect

Resolution

CVBIA

failed to assign to variable '{0}' from binding '{1}': the variable has an incompatible type

{0}	

Effect

Resolution

CVBIE

failed to resolve variable '{0}' on binding '{1}': the variable has enum type {2} and should have enum type {3}

{0}	

Effect

Resolution

CVBIO

failed to resolve variable '{0}' on binding '{1}': the object is a {2} and cannot be converted to a {3}

{0}	

Effect

Resolution

CVBIV

failed to resolve variable '{0}' on binding '{1}': the variable has type {2} and should have type {3}

{0}	

Effect

Resolution

CVBMA

failed to assign to variable '{0}' from binding '{1}': the variable could not be found

{0}	

Effect

Resolution

CVBMV

failed to resolve variable '{0}' on binding '{1}': the variable could not be found

{0}	

Effect

Resolution

CVBROA

failed to assign to variable '{0}' from binding '{1}': the variable is read only

{0}	

Effect

Resolution

CWWIW

unable to watch variable {0} of type {1} - only variable stores can be watched

{0}	

Effect

Resolution

CWWMW

unable to find variable {0} to watch

{0}	

Effect

Resolution

Errors

CCEE

Failed to execute Command '{0}' on '{1}': {2}

{0}	

Effect

Resolution

CCIGPF

Failed to process Node '{0}': the Node yielded a value other than null or IEnumerator

{0}	

Effect

Resolution

CCNMF

failed to set target: unable to find field {0} for instruction graph node {1}

{0}	

Effect

Resolution

CCNMI

failed to set target: index {0} is out of range for instruction graph node {1}

{0}	

Effect

Resolution

CCNMK

failed to set target: unable to find key {0} for instruction graph node {1}

{0}	

Effect

Resolution

CCSONIO

failed to create object for {0}: an object of type '{1}' could not be instantiated

{0}	

Effect

Resolution

CCSONIT

failed to create object for {0}: the type '{1}' could not be found

{0}	

Effect

Resolution

CEEE

Failed to execute Expression '{0}' on '{1}': {2}

{0}	

Effect

Resolution

CELDK

Failed to add keyword '{0}': a keyword with the same text has already been added

{0}	

Effect

Resolution

CELDL

Failed to add constant '{0}': a constant with the same text has already been added

{0}	

Effect

Resolution

CEPDC

Failed to add Command '{0}': a Command with the same name has already been added

{0}	

Effect

Resolution

CEPDIO

Failed to add infix operator '{0}': an infix operator with the same symbol has already been added

{0}	

Effect

Resolution

CEPDPO

Failed to add prefix operator '{0}': a prefix operator with the same symbol has already been added

{0}	

Effect

Resolution

CEPE

Failed to parse Expression at location {1} ({2}): {3} Expression: {0}

{0}	

Effect

Resolution

CEPMC

Failed to remove Command '{0}': a Command with the same name has not been added

{0}	

Effect

Resolution

CETE

Failed to parse Expression at location {1}: {2} Expression: {0}

{0}	

Effect

Resolution

CIAR

Failed to run Instruction '{0}': the Instruction is already running

{0}	

Effect

Resolution

CISIC

failed to create context for {0}: the variable '{1}' does not satisfy the constraint

{0}	

Effect

Resolution

CISII

failed to create input for {0}: the variable '{1}' does not satisfy the constraint

{0}	

Effect

Resolution

CISIOT

failed to store output {0}: the variable '{1}' has an incompatible type

{0}	

Effect

Resolution

CISMI

failed to read input {0}: the variable '{1}' could not be found

{0}	

Effect

Resolution

CISMO

failed to store output {0}: the variable '{1}' could not be found

{0}	

Effect

Resolution

CISROO

failed to store output {0}: the variable '{1}' is read only

{0}	

Effect

Resolution

CMVSIF

failed to map field '{0}' of type '{1}': Only VariableValue types can be mapped

{0}	

Effect

Resolution

CMVSIP

failed to map property '{0}' of type '{1}': Only VariableValue types can be mapped

{0}	

Effect

Resolution

CSCII

Failed to create item {0}: the variable '{1}' is not an IVariableStore or IVariableList

{0}	

Effect

Resolution

CSCMB

Failed to initialize item {0}: the template '{1}' does not have a Binding Root

{0}	

Effect

Resolution

CSCMI

Failed to create item {0}: the variable '{1}' could not be found

{0}	

Effect

Resolution

CSQIS

Unable to run sequence for {0}: index {1} has no connection

{0}	

Effect

Resolution

CTMS

Failed to load Transition {0}: the shader has not been set

{0}	

Effect

Resolution

CVDII

Failed to initialize variable: the definition specifies type {0} but the initializer returned type {1}

{0}	

Effect

Resolution

ISCMC

Failed to create item {0}: SelectionControl '{1}' does not have a child with the specified name

{0}	

Effect

Resolution

ISCMT

Failed to create item {0}: the object template has not been assigned

{0}	

Effect

Resolution