

Unity Composition Reference

PiRho Soft

AbsCommand	1
Description	1
Constructors	1
AcosCommand	1
Description	1
Constructors	1
ActivateInterfaceNode	1
Description	1
Constructors	1
AddAssignOperator	2
Description	2
Constructors	2
AddOperator	2
Description	2
Static Methods	2
AndAssignOperator	2
Description	2
Constructors	3
AndOperator	3
Description	3
Static Methods	3
AnimationClipVariableSource	3
Description	3
Constructors	3
AsinCommand	3
Description	3
Constructors	4
AssignOperator	4
Description	4
Constructors	4
AtanCommand	4
Description	4
Constructors	4
AutoSizeText	5
Description	5
Constructors	5
AxisInput	5
Description	5
Constructors	5
BarBinding	5
Description	5

Constructors	5
BindingAnimationStatus	6
Description	6
Constructors	6
BindingControl	6
Description	6
Constructors	6
BindingUpdater	7
Description	7
Constructors	7
BooleanVariableSource	7
Description	7
Constructors	7
BranchNode	7
Description	7
Constructors	7
BreakNode	8
Description	8
Constructors	8
ButtonInput	8
Description	8
Constructors	8
CeilingCommand	9
Description	9
Constructors	9
ClampCommand	9
Description	9
Constructors	9
ClearTransition	9
Description	9
Constructors	9
Colors	10
Description	10
Static Fields	10
Command	10
Description	10
Static Fields	11
CommandEvaluationException	11
Description	11
Constructors	11
CommandOperation	11

Description	11
Constructors	11
CommandSet	12
Description	12
Constructors	12
CommentNode	12
Description	12
Constructors	12
Composition	13
Description	13
Static Fields	13
CompositionManager	13
Description	13
Constructors	13
ConditionalNode	13
Description	13
Constructors	13
ConnectionData	14
Description	14
Constructors	14
ConstantCommand	15
Description	15
Constructors	15
CosCommand	15
Description	15
Constructors	15
CreateInstructionGraphNodeMenuAttribute	15
Description	15
Constructors	15
CreateObjectNode	16
Description	16
Constructors	16
CustomCommand	16
Description	16
Constructors	16
CustomCommandList	17
Description	17
Constructors	17
Cutoff	17
Description	17
Public Methods	17

DeactivateInterfaceNode	17
Description	17
Constructors	18
DependentObjectList	18
Description	18
Constructors	18
DestroyObjectNode	18
Description	18
Constructors	18
DisableComponentNode	19
Description	19
Constructors	19
DisableGameObjectNode	19
Description	19
Constructors	19
Dissolve	19
Description	20
Constructors	20
DivideAssignOperator	20
Description	20
Constructors	20
DivideOperator	20
Description	20
Static Methods	20
EnableComponentNode	21
Description	21
Constructors	21
EnableGameObjectNode	21
Description	21
Constructors	21
EqualOperator	22
Description	22
Constructors	22
ExponentAssignOperator	22
Description	22
Constructors	22
ExponentOperator	22
Description	22
Static Methods	22
Expression	23
Description	23

Constructors	23
ExpressionCommand	23
Description	23
Static Fields	23
ExpressionCompilationResult	24
Description	24
Public Fields	24
ExpressionEvaluationException	24
Description	24
Constructors	24
ExpressionLexer	24
Description	24
Static Methods	24
ExpressionNode	25
Description	25
Constructors	25
ExpressionParseException	25
Description	25
Constructors	25
ExpressionParser	25
Description	26
Static Methods	26
ExpressionToken	26
Description	26
Constructors	26
ExpressionTokenizeException	26
Description	26
Constructors	26
ExpressionTokenType	27
Description	27
Values	27
Fade	27
Description	27
Constructors	27
FloorCommand	28
Description	28
Constructors	28
FocusIndicator	28
Description	28
Constructors	28
GreaterOperator	28

Description	28
Constructors	28
GreaterOrEqualOperator	29
Description	29
Constructors	29
HideControlNode	29
Description	29
Constructors	29
IndexedVariableStore	29
Description	29
Public Properties	29
ILoopNode	30
Description	30
ImageBinding	31
Description	31
Static Fields	31
IndexedVariableStore	31
Description	31
Static Fields	31
IndexedVariableStore	31
Description	32
Constructors	32
InequalOperator	32
Description	32
Constructors	32
InfixOperation	32
Description	32
Public Fields	32
Instruction	33
Description	33
Public Properties	33
InstructionCaller	33
Description	33
Constructors	33
InstructionContext	34
Description	34
Constructors	34
InstructionGraph	34
Description	34
Static Fields	34
InstructionGraphNode	35

Description	35
Public Fields	35
InstructionGraphNodeDictionary	36
Description	36
Constructors	36
InstructionGraphNodeList	36
Description	36
Constructors	36
InstructionInput	36
Description	36
Constructors	37
InstructionInputType	37
Description	37
Values	37
InstructionNode	37
Description	37
Constructors	37
InstructionOutput	38
Description	38
Constructors	38
InstructionOutputType	38
Description	38
Values	38
InstructionStore	38
Description	38
Static Fields	38
IntegerVariableSource	39
Description	39
Constructors	39
Interface	39
Description	40
Constructors	40
InterfaceBinding	40
Description	40
Static Methods	40
InterfaceControl	41
Description	41
Constructors	41
InterfaceControlDictionary	41
Description	41
Constructors	41

InterfaceManager	41
Description	42
Constructors	42
InterfaceReference	42
Description	42
Constructors	42
InvertOperator	43
Description	43
Constructors	43
ISequenceNode	43
Description	43
ItemSelector	44
Description	44
Constructors	44
IterateNode	44
Description	44
Constructors	44
IVariableList	44
Description	45
Public Properties	45
IVariableListener	45
Description	45
Public Methods	45
IVariableStore	45
Description	45
Public Methods	45
JoinEnumerator	45
Description	45
Static Fields	46
LerpCommand	46
Description	46
Constructors	46
LessOperator	46
Description	46
Constructors	46
LessOrEqualOperator	47
Description	47
Constructors	47
LiteralOperation	47
Description	47
Constructors	47

LocalVariableStore	47
Description	47
Public Methods	47
LogCommand	48
Description	48
Constructors	48
LookupOperation	48
Description	48
Constructors	48
LoopNode	48
Description	48
Constructors	49
MappedVariableStore	49
Description	49
Constructors	49
MaxCommand	49
Description	50
Constructors	50
Message	50
Description	50
Constructors	50
MessageBinding	50
Description	50
Constructors	50
MessageControl	51
Description	51
Constructors	51
MessageControlDisplay	51
Description	51
Values	51
MessageInteractionType	52
Description	52
Values	52
MessageNode	52
Description	52
Constructors	52
MinCommand	53
Description	53
Constructors	53
MockupConnection	53
Description	53

Constructors	53
MockupConnectionList	53
Description	53
Constructors	53
MockupGraph	53
Description	54
Constructors	54
MockupNode	54
Description	54
Constructors	54
ModuloAssignOperator	54
Description	55
Constructors	55
ModuloOperator	55
Description	55
Static Methods	55
MultiplyAssignOperator	55
Description	55
Constructors	55
MultiplyOperator	55
Description	56
Static Methods	56
NegateOperator	56
Description	56
Constructors	56
NodeData	56
Description	56
Static Fields	56
NumberVariableSource	57
Description	57
Constructors	57
ObjectSource	57
Description	57
Values	57
ObjectVariableSource	58
Description	58
Constructors	58
Operation	58
Description	58
Static Fields	58
OperatorPrecedence	58

Description	59
Static Fields	59
OrAssignOperator	59
Description	59
Constructors	59
OrOperator	59
Description	59
Static Methods	60
ParameterStore	60
Description	60
Static Fields	60
Pixelate	60
Description	60
Constructors	60
PlayAnimation	61
Description	61
Constructors	61
PlayAnimationState	61
Description	61
Constructors	61
PlaybackState	62
Description	62
Values	62
PlayTransition	62
Description	62
Constructors	62
PowCommand	63
Description	63
Constructors	63
PrefixOperation	63
Description	63
Public Fields	63
PrimaryAxis	64
Description	64
Values	64
Property	64
Description	64
Constructors	64
PropertyList	64
Description	64
Constructors	64

PropertyMap	65
Description	65
Public Properties	65
PropertyMap	65
Description	65
Constructors	65
RandomCommand	66
Description	66
Constructors	66
ReadOnlyStore	66
Description	66
Constructors	66
RoundCommand	66
Description	66
Constructors	66
SceneVariableStore	67
Description	67
Constructors	67
ScopedGraph	67
Description	67
Constructors	67
ScrollSelection	67
Description	67
Constructors	67
SelectionBinding	68
Description	68
Constructors	68
SelectionControl	68
Description	68
Constructors	68
SelectionItem	70
Description	70
Constructors	70
SelectionNode	71
Description	71
Constructors	71
SelectionNodeItem	71
Description	71
Constructors	71
SelectionNodeItemList	71
Description	72

Constructors	72
SequenceNode	72
Description	72
Constructors	72
SerializedVariable	72
Description	72
Constructors	72
SetAnimationParameter	73
Description	73
Constructors	73
SetVariableResult	73
Description	73
Values	73
ShowControlNode	74
Description	74
Constructors	74
SignCommand	74
Description	74
Constructors	74
SimpleGraph	74
Description	75
Constructors	75
SinCommand	75
Description	75
Constructors	75
SqrtCommand	75
Description	75
Constructors	75
StringVariableSource	76
Description	76
Constructors	76
SubtractAssignOperator	76
Description	76
Constructors	76
SubtractOperator	76
Description	76
Static Methods	76
TanCommand	77
Description	77
Constructors	77
TernaryOperator	77

Description	77
Constructors	77
TextBinding	77
Description	77
Constructors	77
Transition	78
Description	78
Public Fields	78
TransitionList	78
Description	78
Constructors	78
TransitionManager	79
Description	79
Constructors	79
TransitionPhase	79
Description	79
Values	79
TransitionRenderer	79
Description	79
Constructors	79
TransitionVariableSource	80
Description	80
Constructors	80
TruncateCommand	80
Description	80
Constructors	80
UpdateBindingNode	80
Description	80
Constructors	80
Variable	81
Description	81
Static Properties	81
VariableAvailabilitiesAttribute	81
Description	81
Constructors	81
VariableDefinition	81
Description	82
Static Fields	82
VariableInitializerAttribute	82
Description	82
Constructors	82

VariableInitializerType	83
Description	83
Values	83
VariableList	83
Description	83
Constructors	83
VariableListener	84
Description	84
Constructors	84
VariableMap	84
Description	84
Constructors	84
VariableReference	85
Description	85
Constructors	85
VariableSchema	85
Description	85
Constructors	85
VariableSource	86
Description	86
Public Fields	86
VariableSourceType	86
Description	86
Values	86
VariableStore	86
Description	86
Constructors	86
VariableType	87
Description	87
Values	87
VariableValue	87
Description	87
Static Fields	88
VisibleBinding	89
Description	89
Constructors	89
WaitNode	89
Description	89
Constructors	89
WritableStore	90
Description	90

AbsCommand

PiRhoSoft.CompositionEngine.AbsCommand : [Command](#)

Description

Constructors

AbsCommand()

Public Methods

VariableValue Evaluate(**IVariableStore** variables, **string** name, **List<Operation>** parameters)
(virtual)

<<<

AcosCommand

PiRhoSoft.CompositionEngine.AcosCommand : [Command](#)

Description

Constructors

AcosCommand()

Public Methods

VariableValue Evaluate(**IVariableStore** variables, **string** name, **List<Operation>** parameters)
(virtual)

<<<

ActivateInterfaceNode

PiRhoSoft.CompositionEngine.ActivateInterfaceNode : [InstructionGraphNode](#)

Description

Constructors

ActivateInterfaceNode()

Public Fields

InstructionGraphNode Next

string InterfaceName

Public Properties

Color *NodeColor (read only) (virtual)*

Protected Methods

IEnumerator **Run_**(**InstructionGraph** graph, **InstructionStore** variables, **int** iteration) (virtual)

<<<

AddAssignOperator

PiRhoSoft.CompositionEngine.AddAssignOperator : [AssignOperator](#)

Description

Constructors

AddAssignOperator()

Public Methods

VariableValue **Evaluate**(**IVariableStore** variables) (virtual)

<<<

AddOperator

PiRhoSoft.CompositionEngine.AddOperator : [InfixOperation](#)

Description

Static Methods

VariableValue **Add**(**Operation** expression, **VariableValue** left (ref), **VariableValue** right (ref))

Constructors

AddOperator()

Public Methods

VariableValue **Evaluate**(**IVariableStore** variables) (virtual)

<<<

AndAssignOperator

PiRhoSoft.CompositionEngine.AndAssignOperator : [AssignOperator](#)

Description

Constructors

AndAssignOperator()

Public Methods

VariableValue Evaluate(**IVariableStore** variables) (virtual)

<<<

AndOperator

PiRhoSoft.CompositionEngine.AndOperator : [InfixOperation](#)

Description

Static Methods

VariableValue And(**Operation** expression, **VariableValue** left (ref), **VariableValue** right (ref))

Constructors

AndOperator()

Public Methods

VariableValue Evaluate(**IVariableStore** variables) (virtual)

<<<

AnimationClipVariableSource

PiRhoSoft.CompositionEngine.AnimationClipVariableSource : [ObjectVariableSource](#)<
[AnimationClip](#)>

Description

Constructors

AnimationClipVariableSource()

<<<

AsinCommand

PiRhoSoft.CompositionEngine.AsinCommand : [Command](#)

Description

Constructors

AsinCommand()

Public Methods

VariableValue Evaluate(**IVariableStore** variables, **string** name, **List<Operation>** parameters) (virtual)

<<<

AssignOperator

PiRhoSoft.CompositionEngine.AssignOperator : [InfixOperation](#)

Description

Constructors

AssignOperator()

Public Methods

VariableValue Evaluate(**IVariableStore** variables) (virtual)

void GetInputs(**List<VariableDefinition>** inputs, **string** source) (virtual)

void GetOutputs(**List<VariableDefinition>** outputs, **string** source) (virtual)

Protected Methods

VariableValue Assign(**IVariableStore** variables, **VariableValue** value (ref))

<<<

AtanCommand

PiRhoSoft.CompositionEngine.AtanCommand : [Command](#)

Description

Constructors

AtanCommand()

Public Methods

VariableValue Evaluate(**IVariableStore** variables, **string** name, **List<Operation>** parameters) (virtual)

<<<

AutoSizeText

PiRhoSoft.CompositionEngine.AutoSizeText : [MonoBehaviour](#)

Description

Constructors

AutoSizeText()

<<<

AxisInput

PiRhoSoft.CompositionEngine.AxisInput : [MonoBehaviour](#), [IEventSystemHandler](#),
[IPointerDownHandler](#), [IPointerUpHandler](#)

Description

Constructors

AxisInput()

Public Fields

string *AxisName*

float *AxisValue*

Public Methods

void OnPointerDown([PointerEventData](#) eventData) (virtual)

void OnPointerUp([PointerEventData](#) eventData) (virtual)

<<<

BarBinding

PiRhoSoft.CompositionEngine.BarBinding : [InterfaceBinding](#)

Description

Constructors

BarBinding()

Public Fields

float *Speed*

VariableReference *AmountVariable*

VariableReference *TotalVariable*

Gradient *FillColors*

Public Methods

void **UpdateBinding**(**IVariableStore** *variables*, **BindingAnimationStatus** *status*) (*virtual*)

Protected Fields

Image *_image*

Protected Methods

float **GetFill**(**IVariableStore** *variables*)

<<<

BindingAnimationStatus

PiRhoSoft.CompositionEngine.BindingAnimationStatus

Description

Constructors

BindingAnimationStatus()

Public Methods

void **Reset()**

bool **IsFinished()**

void **Increment()**

void **Decrement()**

<<<

BindingControl

PiRhoSoft.CompositionEngine.BindingControl : [InterfaceControl](#)

Description

Constructors

BindingControl()

Public Fields

VariableReference *Binding*

Public Methods

```
void UpdateBindings(IVariableStore variables, string group, BindingAnimationStatus status)
(virtual)
<<<
```

BindingUpdater

PiRhoSoft.CompositionEngine.BindingUpdater : [MonoBehaviour](#)

Description

Constructors

BindingUpdater()

Public Fields

string *Group*

<<<

BooleanVariableSource

PiRhoSoft.CompositionEngine.BooleanVariableSource : [VariableSource](#)<bool>

Description

Constructors

BooleanVariableSource()

BooleanVariableSource(bool *defaultValue***)**

Protected Methods

bool TryGetValue([VariableValue](#) variable, **bool** value (out)) (virtual)

<<<

BranchNode

PiRhoSoft.CompositionEngine.BranchNode : [InstructionGraphNode](#)

Description

Constructors

BranchNode()

Public Fields

Expression *Switch*

InstructionGraphNodeDictionary *Outputs*

InstructionGraphNode *Default*

Public Properties

Color *NodeColor (read only) (virtual)*

Public Methods

void GetInputs(List<VariableDefinition> inputs) (virtual)

Protected Methods

IEnumerator Run_(InstructionGraph graph, InstructionStore variables, int iteration) (virtual)

<<<

BreakNode

PiRhoSoft.CompositionEngine.BreakNode : [InstructionGraphNode](#)

Description

Constructors

BreakNode()

Public Properties

Color *NodeColor (read only) (virtual)*

Protected Methods

IEnumerator Run_(InstructionGraph graph, InstructionStore variables, int iteration) (virtual)

<<<

ButtonInput

PiRhoSoft.CompositionEngine.ButtonInput : [MonoBehaviour](#), [IEventSystemHandler](#),
[IPointerDownHandler](#), [IPointerUpHandler](#)

Description

Constructors

ButtonInput()

Public Fields

string ButtonName

Public Methods

void OnPointerDown([PointerEventData](#) eventData) (virtual)

void OnPointerUp([PointerEventData](#) eventData) (virtual)

<<<

CeilingCommand

PiRhoSoft.CompositionEngine.CeilingCommand : [Command](#)

Description

Constructors

CeilingCommand()

Public Methods

[VariableValue](#) Evaluate([IVariableStore](#) variables, string name, [List<Operation>](#) parameters) (virtual)

<<<

ClampCommand

PiRhoSoft.CompositionEngine.ClampCommand : [Command](#)

Description

Constructors

ClampCommand()

Public Methods

[VariableValue](#) Evaluate([IVariableStore](#) variables, string name, [List<Operation>](#) parameters) (virtual)

<<<

ClearTransition

PiRhoSoft.CompositionEngine.ClearTransition : [InstructionGraphNode](#)

Description

Constructors

ClearTransition()

Public Fields

InstructionGraphNode *Next*

Public Properties

Color *NodeColor (read only) (virtual)*

Protected Methods

IEnumerator **Run_**(**InstructionGraph** graph, **InstructionStore** variables, **int** iteration) (virtual)

<<<

Colors

PiRhoSoft.CompositionEngine.Colors

Description

Static Fields

Color *Start*

Color *Default*

Color *ExecutionLight*

Color *ExecutionDark*

Color *Animation*

Color *Sequence*

Color *Loop*

Color *Branch*

Color *Break*

Color *Sequencing*

Color *SequencingLight*

Color *SequencingDark*

Color *Interface*

Color *InterfaceLight*

Color *InterfaceDark*

Color *InterfaceCyan*

Color *InterfaceTeal*

<<<

Command

PiRhoSoft.CompositionEngine.Command

Description

Static Fields

```
string WrongParameterCountException
string WrongParameterRangeException
string TooFewParametersException
string TooManyParametersException
string WrongParameterType1Exception
string WrongParameterType2Exception
string InvalidRangeException
    # Public Methods
```

```
VariableValue Evaluate(IVariableStore variables, string name, List<Operation> parameters)
(abstract)
    # Protected Constructors
```

```
Command()
<<<
```

CommandEvaluationException

PiRhoSoft.CompositionEngine.CommandEvaluationException : Exception

Description

Constructors

```
CommandEvaluationException(string command, string error)
CommandEvaluationException(string command, string errorFormat, Object[] arguments)
    # Public Fields

string Command
<<<
```

CommandOperation

PiRhoSoft.CompositionEngine.CommandOperation : [Operation](#)

Description

Constructors

```
CommandOperation(string name, List<Operation> parameters)
    # Public Properties
```

string *Name (read only)*

List<[Operation](#)> *Parameters (read only)*

Public Methods

void ToString(**StringBuilder** *builder*) (virtual)

void GetInputs(**List**<[VariableDefinition](#)> *inputs*, **string** *source*) (virtual)

VariableValue **Evaluate**(**IVariableStore** *variables*) (virtual)

<<<

CommandSet

PiRhoSoft.CompositionEngine.CommandSet : [ScriptableObject](#)

Description

Constructors

CommandSet()

Public Fields

CustomCommandList *CustomCommands*

Public Methods

void AddExpression(**string** *name*)

void RemoveCommand(**int** *index*)

void SetExpression(**int** *index*, [Expression](#) *expression*)

bool IsNameAvailable(**string** *name*)

Protected Methods

void RegisterCommands() (virtual)

void UnregisterCommands() (virtual)

<<<

CommentNode

PiRhoSoft.CompositionEngine.CommentNode : [InstructionGraphNode](#)

Description

Constructors

CommentNode()

Public Fields

string *Comment*

Protected Methods

IEnumerator **Run_**(**InstructionGraph** *graph*, **InstructionStore** *variables*, **int** *iteration*) (virtual)

<<<

Composition

PiRhoSoft.CompositionEngine.Composition

Description

Static Fields

string *DocumentationUrl*

<<<

CompositionManager

PiRhoSoft.CompositionEngine.CompositionManager : [SingletonBehaviour](#)<[CompositionManager](#)>

Description

Constructors

CompositionManager()

Public Fields

CommandSet *Commands*

Public Methods

void **RunInstruction**(**Instruction** *instruction*, **InstructionContext** *context*, **Object** *thisObject*)

void **RunInstruction**(**InstructionCaller** *caller*, **InstructionContext** *context*, **Object** *thisObject*)

<<<

ConditionalNode

PiRhoSoft.CompositionEngine.ConditionalNode : [InstructionGraphNode](#)

Description

Constructors

ConditionalNode()

Public Fields

InstructionGraphNode *OnTrue*

InstructionGraphNode *OnFalse*

Expression *Condition*

Public Properties

Color *NodeColor (read only) (virtual)*

Public Methods

void GetInputs(List<VariableDefinition> inputs) (virtual)

Protected Methods

IEnumerator Run_(InstructionGraph graph, InstructionStore variables, int iteration) (virtual)

<<<

ConnectionData

PiRhoSoft.CompositionEngine.ConnectionData

Description

Constructors

ConnectionData(string field, string key, int index, InstructionGraphNode from, InstructionGraphNode to, int fromIndex)

Public Properties

string Field *(read only)*

string FieldKey *(read only)*

int FieldIndex *(read only)*

InstructionGraphNode From *(read only)*

int FromIndex *(read only)*

InstructionGraphNode To *(read only)*

NodeData Target *(read only)*

string Name *(read only)*

Public Methods

bool Equals(Object obj) (virtual)

int GetHashCode() (virtual)

void SetTarget(NodeData target)

void ChangeTarget(NodeData target)

void ApplyConnection(Object obj, InstructionGraphNode target)

<<<

ConstantCommand

PiRhoSoft.CompositionEngine.ConstantCommand : [Command](#)

Description

Constructors

ConstantCommand([VariableValue](#) value)

Public Properties

[VariableValue](#) Value (read only)

Public Methods

[VariableValue](#) Evaluate([IVariableStore](#) variables, string name, [List](#)<[Operation](#)> parameters)
(virtual)

<<<

CosCommand

PiRhoSoft.CompositionEngine.CosCommand : [Command](#)

Description

Constructors

CosCommand()

Public Methods

[VariableValue](#) Evaluate([IVariableStore](#) variables, string name, [List](#)<[Operation](#)> parameters)
(virtual)

<<<

CreateInstructionGraphNodeMenuAttribute

PiRhoSoft.CompositionEngine.CreateInstructionGraphNodeMenuAttribute : Attribute

Description

Constructors

CreateInstructionGraphNodeMenuAttribute(string menuName, int order)

Public Properties

string *MenuName (read only)*

int *Order (read only)*

<<<

CreateObjectNode

PiRhoSoft.CompositionEngine.CreateObjectNode : [InstructionGraphNode](#)

Description

Constructors

CreateObjectNode()

Public Fields

[InstructionGraphNode](#) *Next*

[GameObject](#) *Prefab*

string *ObjectName*

[Vector2](#) *Position*

[VariableReference](#) *Parent*

Public Properties

[Color](#) *NodeColor (read only) (virtual)*

Public Methods

void **GetInputs**([List](#)<[VariableDefinition](#)> *inputs*) *(virtual)*

Protected Methods

[IEnumerator](#) **Run_**([InstructionGraph](#) *graph*, [InstructionStore](#) *variables*, **int** *iteration*) *(virtual)*

<<<

CustomCommand

PiRhoSoft.CompositionEngine.CustomCommand

Description

Constructors

CustomCommand()

Public Fields

string *Name*

[Expression](#) *Expression*

<<<

CustomCommandList

PiRhoSoft.CompositionEngine.CustomCommandList : [SerializedList](#)<[CustomCommand](#)>

Description

Constructors

CustomCommandList()

<<<

Cutoff

PiRhoSoft.CompositionEngine.Cutoff : [Transition](#)

Description

Public Methods

void Process(float time, [TransitionPhase](#) phase) (virtual)

void Render([TransitionRenderer](#) renderer, [RenderTexture](#) source, [RenderTexture](#) destination)
(virtual)

Protected Constructors

Cutoff()

Protected Methods

void SetTexture([Texture2D](#) texture)

void SetColor([Color](#) color)

void SetCutoff(float cutoff)

void SetFade(float fade)

void SetDistort(bool distort)

void Setup() (virtual)

void Update() (virtual)

<<<

DeactivateInterfaceNode

PiRhoSoft.CompositionEngine.DeactivateInterfaceNode : [InstructionGraphNode](#)

Description

Constructors

DeactivateInterfaceNode()

Public Fields

InstructionGraphNode *Next*

string *InterfaceName*

Public Properties

Color *NodeColor (read only) (virtual)*

Protected Methods

IEnumerator **Run_**(**InstructionGraph** *graph*, **InstructionStore** *variables*, **int** *iteration*) (virtual)

<<<

DependentObjectList

PiRhoSoft.CompositionEngine.DependentObjectList : [SerializedList<GameObject>](#)

Description

Constructors

DependentObjectList()

<<<

DestroyObjectNode

PiRhoSoft.CompositionEngine.DestroyObjectNode : [InstructionGraphNode](#)

Description

Constructors

DestroyObjectNode()

Public Fields

InstructionGraphNode *Next*

Public Properties

Color *NodeColor (read only) (virtual)*

Protected Methods

IEnumerator **Run_**(**InstructionGraph** *graph*, **InstructionStore** *variables*, **int** *iteration*) (virtual)

<<<

DisableComponentNode

PiRhoSoft.CompositionEngine.DisableComponentNode : [InstructionGraphNode](#)

Description

Constructors

DisableComponentNode()

Public Fields

[InstructionGraphNode](#) *Next*

Public Properties

[Color](#) *NodeColor (read only) (virtual)*

Protected Methods

[IEnumerator](#) **Run_**([InstructionGraph](#) graph, [InstructionStore](#) variables, **int** iteration) (virtual)

<<<

DisableGameObjectNode

PiRhoSoft.CompositionEngine.DisableGameObjectNode : [InstructionGraphNode](#)

Description

Constructors

DisableGameObjectNode()

Public Fields

[InstructionGraphNode](#) *Next*

Public Properties

[Color](#) *NodeColor (read only) (virtual)*

Protected Methods

[IEnumerator](#) **Run_**([InstructionGraph](#) graph, [InstructionStore](#) variables, **int** iteration) (virtual)

<<<

Dissolve

PiRhoSoft.CompositionEngine.Dissolve : [Cutoff](#)

Description

Constructors

Dissolve()

Public Fields

Color *Color*

Texture2D *Texture*

Vector2Int *TextureSize*

float *NoiseScale*

Public Methods

void Begin(**TransitionPhase** *phase*) (*virtual*)

void End() (*virtual*)

void Process(**float** *time*, **TransitionPhase** *phase*) (*virtual*)

Protected Methods

void Setup() (*virtual*)

<<<

DivideAssignOperator

PiRhoSoft.CompositionEngine.DivideAssignOperator : [AssignOperator](#)

Description

Constructors

DivideAssignOperator()

Public Methods

VariableValue **Evaluate**(**IVariableStore** *variables*) (*virtual*)

<<<

DivideOperator

PiRhoSoft.CompositionEngine.DivideOperator : [InfixOperation](#)

Description

Static Methods

VariableValue **Divide**(**VariableValue** *left (ref)*, **VariableValue** *right (ref)*)

Constructors

DivideOperator()

Public Methods

VariableValue Evaluate(**IVariableStore** variables) (virtual)

<<<

EnableComponentNode

PiRhoSoft.CompositionEngine.EnableComponentNode : [InstructionGraphNode](#)

Description

Constructors

EnableComponentNode()

Public Fields

InstructionGraphNode Next

Public Properties

Color NodeColor (read only) (virtual)

Protected Methods

IEnumerator Run_(**InstructionGraph** graph, **InstructionStore** variables, **int** iteration) (virtual)

<<<

EnableGameObjectNode

PiRhoSoft.CompositionEngine.EnableGameObjectNode : [InstructionGraphNode](#)

Description

Constructors

EnableGameObjectNode()

Public Fields

InstructionGraphNode Next

Public Properties

Color NodeColor (read only) (virtual)

Protected Methods

IEnumerator Run_(**InstructionGraph** graph, **InstructionStore** variables, **int** iteration) (virtual)

<<<

EqualOperator

PiRhoSoft.CompositionEngine.EqualOperator : [InfixOperation](#)

Description

Constructors

EqualOperator()

Public Methods

VariableValue Evaluate(**IVariableStore** variables) (virtual)

<<<

ExponentAssignOperator

PiRhoSoft.CompositionEngine.ExponentAssignOperator : [AssignOperator](#)

Description

Constructors

ExponentAssignOperator()

Public Methods

VariableValue Evaluate(**IVariableStore** variables) (virtual)

<<<

ExponentOperator

PiRhoSoft.CompositionEngine.ExponentOperator : [InfixOperation](#)

Description

Static Methods

VariableValue Raise(**VariableValue** left (ref), **VariableValue** right (ref))

Constructors

ExponentOperator()

Public Methods

VariableValue Evaluate(**IVariableStore** variables) (virtual)

<<<

Expression

PiRhoSoft.CompositionEngine.Expression

Description

Constructors

Expression()

Public Fields

bool *IsExpanded*

Public Properties

bool *IsValid (read only)*

bool *HasError (read only)*

string *Statement (read only)*

Public Methods

ExpressionCompilationResult **SetStatement**(string *statement*)

void **GetInputs**(List<**VariableDefinition**> *inputs*, string *source*)

void **GetOutputs**(List<**VariableDefinition**> *outputs*, string *source*)

VariableValue **Execute**(Object *context*, **IVariableStore** *variables*)

VariableValue **Execute**(Object *context*, **IVariableStore** *variables*, **VariableType** *expectedType*)

VariableValue **Evaluate**(**IVariableStore** *variables*)

<<<

ExpressionCommand

PiRhoSoft.CompositionEngine.ExpressionCommand : [Command](#)

Description

Static Fields

int *InitialStoreCount*

Stack<**ParameterStore**> *Stores*

Static Methods

ParameterStore **ReserveStore**()

void **ReleaseStore**(**ParameterStore** *store*)

Constructors

ExpressionCommand(**Expression** *expression*)

Public Properties

Expression *Expression (read only)*

Public Methods

VariableValue **Evaluate**(**IVariableStore** *variables*, **string** *name*, **List**<**Operation**> *parameters*)
(*virtual*)

<<<

ExpressionCompilationResult

PiRhoSoft.CompositionEngine.ExpressionCompilationResult : ValueType

Description

Public Fields

bool *Success*

string *Message*

<<<

ExpressionEvaluationException

PiRhoSoft.CompositionEngine.ExpressionEvaluationException : Exception

Description

Constructors

ExpressionEvaluationException(**string** *error*)

ExpressionEvaluationException(**string** *errorFormat*, **Object**[] *arguments*)

<<<

ExpressionLexer

PiRhoSoft.CompositionEngine.ExpressionLexer

Description

Static Methods

List<**ExpressionToken**> **Tokenize**(**string** *input*)

<<<

ExpressionNode

PiRhoSoft.CompositionEngine.ExpressionNode : [InstructionGraphNode](#)

Description

Constructors

ExpressionNode()

Public Fields

[InstructionGraphNode](#) *Next*

[Expression](#) *Expression*

Public Properties

[Color](#) *NodeColor (read only) (virtual)*

Public Methods

void GetInputs(List<[VariableDefinition](#)> inputs) (virtual)

void GetOutputs(List<[VariableDefinition](#)> outputs) (virtual)

Protected Methods

IEnumerator Run_([InstructionGraph](#) graph, [InstructionStore](#) variables, int iteration) (virtual)

<<<

ExpressionParseException

PiRhoSoft.CompositionEngine.ExpressionParseException : Exception

Description

Constructors

ExpressionParseException(int location, string error)

ExpressionParseException(int location, string errorFormat, Object[] arguments)

Public Fields

int Location

<<<

ExpressionParser

PiRhoSoft.CompositionEngine.ExpressionParser

Description

Static Methods

```
void AddCommand(string name, Command command)
void RemoveCommand(string name)
Command GetCommand(string name)
void AddPrefixOperator<OperatorType>(string symbol)
void AddInfixOperator<OperatorType>(string symbol, OperatorPrecedence precedence)
void AddTernaryOperator(string symbol, OperatorPrecedence precedence)
List<Operation> Parse(List<ExpressionToken> tokens)
<<<
```

ExpressionToken

PiRhoSoft.CompositionEngine.ExpressionToken

Description

Constructors

```
ExpressionToken()
    # Public Fields

int Location
ExpressionTokenType Type
string Text
int Integer
float Number
<<<
```

ExpressionTokenizeException

PiRhoSoft.CompositionEngine.ExpressionTokenizeException : Exception

Description

Constructors

```
ExpressionTokenizeException(int location, string error)
ExpressionTokenizeException(int location, string errorFormat, Object[] arguments)
    # Public Fields
```

int *Location*

<<<

ExpressionTokenType

PiRhoSoft.CompositionEngine.ExpressionTokenType

Description

Values

ExpressionTokenType *Sentinel*

ExpressionTokenType *Boolean*

ExpressionTokenType *Integer*

ExpressionTokenType *Number*

ExpressionTokenType *String*

ExpressionTokenType *Null*

ExpressionTokenType *Identifier*

ExpressionTokenType *Command*

ExpressionTokenType *Operator*

ExpressionTokenType *StartGroup*

ExpressionTokenType *EndGroup*

ExpressionTokenType *Separator*

<<<

Fade

PiRhoSoft.CompositionEngine.Fade : **Cutoff**

Description

Constructors

Fade()

Public Fields

Color *Color*

Protected Methods

void Setup() (*virtual*)

<<<

FloorCommand

PiRhoSoft.CompositionEngine.FloorCommand : [Command](#)

Description

Constructors

FloorCommand()

Public Methods

VariableValue Evaluate(**IVariableStore** variables, string name, **List<Operation>** parameters) (virtual)

<<<

FocusIndicator

PiRhoSoft.CompositionEngine.FocusIndicator : [MonoBehaviour](#)

Description

Constructors

FocusIndicator()

Public Methods

void SetFocused(bool focused) (virtual)

<<<

GreaterOperator

PiRhoSoft.CompositionEngine.GreaterOperator : [InfixOperation](#)

Description

Constructors

GreaterOperator()

Public Methods

VariableValue Evaluate(**IVariableStore** variables) (virtual)

<<<

GreaterOrEqualOperator

PiRhoSoft.CompositionEngine.GreaterOrEqualOperator : [InfixOperation](#)

Description

Constructors

GreaterOrEqualOperator()

Public Methods

VariableValue Evaluate(**IVariableStore** variables) (virtual)

<<<

HideControlNode

PiRhoSoft.CompositionEngine.HideControlNode : [InstructionGraphNode](#)

Description

Constructors

HideControlNode()

Public Fields

InstructionGraphNode Next

InterfaceReference Control

Public Properties

Color NodeColor (read only) (virtual)

Protected Methods

IEnumerator Run_(**InstructionGraph** graph, **InstructionStore** variables, **int** iteration) (virtual)

<<<

IIndexedVariableStore

PiRhoSoft.CompositionEngine.IIndexedVariableStore : [IVariableStore](#)

Description

Public Properties

int Count (read only) (abstract)

Public Methods

IVariableStore **GetItem(int index)** (*abstract*)

<<<

ILoopNode

PiRhoSoft.CompositionEngine.ILoopNode

Description

ImageBinding

PiRhoSoft.CompositionEngine.ImageBinding : [InterfaceBinding](#)

Description

Static Fields

string *_invalidVariableError*

string *_missingVariableError*

Constructors

ImageBinding()

Public Fields

VariableReference *Variable*

Public Methods

void **UpdateBinding**(**IVariableStore** *variables*, **BindingAnimationStatus** *status*) (virtual)

<<<

IndexedVariableStore

PiRhoSoft.CompositionEngine.IndexedVariableStore

Description

Static Fields

string *ItemVariable*

Static Methods

VariableValue **GetVariable**(**IIndexedVariableStore** *variables*, **string** *name*)

SetVariableResult **SetVariable**(**IIndexedVariableStore** *variables*, **string** *name*, **VariableValue** *value*)

IEnumerable<**string**> **GetVariableNames**(**IIndexedVariableStore** *variables*)

<<<

IndexedVariableStore

PiRhoSoft.CompositionEngine.IndexedVariableStore<*ItemType*> : [SerializedList](#)<*ItemType*>, [IVariableStore](#), [IndexedVariableStore](#)

Description

Constructors

IndexedVariableStore()

Public Methods

IVariableStore **GetItem**(int *index*) (virtual)

VariableValue **GetVariable**(string *name*) (virtual)

SetVariableResult **SetVariable**(string *name*, **VariableValue** *value*) (virtual)

IEnumerable<string> **GetVariableNames**() (virtual)

<<<

InequalOperator

PiRhoSoft.CompositionEngine.InequalOperator : [InfixOperation](#)

Description

Constructors

InequalOperator()

Public Methods

VariableValue **Evaluate**(**IVariableStore** *variables*) (virtual)

<<<

InfixOperation

PiRhoSoft.CompositionEngine.InfixOperation : [Operation](#)

Description

Public Fields

Operation *Left*

string *Symbol*

Operation *Right*

Public Methods

void **ToString**(StringBuilder *builder*) (virtual)

void **GetInputs**(List<**VariableDefinition**> *inputs*, string *source*) (virtual)

Protected Constructors

InfixOperation()

<<<

Instruction

PiRhoSoft.CompositionEngine.Instruction : [ScriptableObject](#)

Description

Public Properties

bool *IsRunning* (read only)

Public Methods

IEnumerator **Execute**(**InstructionStore** variables)

void **GetInputs**(**List**<**VariableDefinition**> inputs) (virtual)

void **GetOutputs**(**List**<**VariableDefinition**> outputs) (virtual)

Protected Constructors

Instruction()

Protected Methods

void **OnEnable**() (virtual)

void **OnDisable**() (virtual)

IEnumerator **Run**(**InstructionStore** variables) (abstract)

<<<

InstructionCaller

PiRhoSoft.CompositionEngine.InstructionCaller

Description

Constructors

InstructionCaller()

Public Properties

Instruction *Instruction*

IList<**InstructionInput**> *Inputs* (read only)

IList<**InstructionOutput**> *Outputs* (read only)

bool *IsRunning* (read only)

Public Methods

IEnumerator Execute(**InstructionContext** context, **Object** thisObject)

void UpdateVariables()

<<<

InstructionContext

PiRhoSoft.CompositionEngine.InstructionContext

Description

Constructors

InstructionContext()

Public Properties

Dictionary<**string**, **string**> Stores (read only)

Public Methods

void SetStore(**string** name, **IVariableStore** store)

void Clear()

<<<

InstructionGraph

PiRhoSoft.CompositionEngine.InstructionGraph : [Instruction](#)

Description

Static Fields

string _processFailedError

bool IsDebugBreakEnabled

bool IsDebugLoggingEnabled

Action<[InstructionGraph](#), [InstructionGraph](#)> OnBreakpointHit

Public Fields

Vector2 StartPosition

Public Properties

InstructionGraphNodeList Nodes (read only)

InstructionStore Store (read only)

PlaybackState DebugState (read only)

bool CanDebugPlay (read only)

bool CanDebugPause (read only)

```

bool CanDebugStep (read only)
bool CanDebugStop (read only)
    # Public Methods

void GetInputs(List<VariableDefinition> inputs) (virtual)
void GetOutputs(List<VariableDefinition> outputs) (virtual)
void GoTo(InstructionGraphNode node, Object thisObject, string name)
void GoTo(InstructionGraphNode node, Object thisObject, string name, int index)
void GoTo(InstructionGraphNode node, Object thisObject, string name, string key)
void Break()
void DebugPlay()
void DebugPause()
void DebugStep()
void DebugStop()
int IsInCallStack(InstructionGraphNode node)
bool IsInCallStack(InstructionGraphNode node, string source)
bool IsExecuting(InstructionGraphNode node)
void GetConnections(NodeData data) (virtual)
void SetConnection(ConnectionData connection, InstructionGraphNode target) (virtual)
    # Protected Constructors

InstructionGraph()
    # Protected Methods

IEnumerator Run(InstructionStore variables, InstructionGraphNode root, string source)
    <<<

```

InstructionGraphNode

PiRhoSoft.CompositionEngine.InstructionGraphNode : [ScriptableObject](#)

Description

Public Fields

```

string Name
VariableReference This
Vector2 GraphPosition
bool IsBreakpoint
    # Public Properties

Color NodeColor (read only) (virtual)
    # Public Methods

```

```

void GetInputs(List<VariableDefinition> inputs) (virtual)
void GetOutputs(List<VariableDefinition> outputs) (virtual)
IEnumerator Run(InstructionGraph graph, InstructionStore variables, int executionIndex)
void GetConnections(NodeData data) (virtual)
void SetConnection(ConnectionData connection, InstructionGraphNode target) (virtual)
    # Protected Constructors

InstructionGraphNode()
    # Protected Methods

IEnumerator Run_(InstructionGraph graph, InstructionStore variables, int iteration) (abstract)
<<<

```

InstructionGraphNodeDictionary

PiRhoSoft.CompositionEngine.InstructionGraphNodeDictionary : SerializedDictionary<string, string>

Description

Constructors

```

InstructionGraphNodeDictionary()
<<<

```

InstructionGraphNodeList

PiRhoSoft.CompositionEngine.InstructionGraphNodeList : SerializedList<InstructionGraphNode>

Description

Constructors

```

InstructionGraphNodeList()
<<<

```

InstructionInput

PiRhoSoft.CompositionEngine.InstructionInput

Description

Constructors

InstructionInput()

Public Fields

[InstructionInputType](#) *Type*

[VariableDefinition](#) *Definition*

[VariableReference](#) *Reference*

[VariableValue](#) *Value*

<<<

InstructionInputType

PiRhoSoft.CompositionEngine.InstructionInputType

Description

Values

[InstructionInputType](#) *Reference*

[InstructionInputType](#) *Value*

<<<

InstructionNode

PiRhoSoft.CompositionEngine.InstructionNode : [InstructionGraphNode](#)

Description

Constructors

InstructionNode()

Public Fields

[InstructionCaller](#) *Instruction*

[InstructionGraphNode](#) *Next*

bool *WaitForCompletion*

Public Properties

Color *NodeColor (read only) (virtual)*

Protected Methods

IEnumerator **Run_**([InstructionGraph](#) graph, [InstructionStore](#) variables, **int** iteration) (virtual)

<<<

InstructionOutput

PiRhoSoft.CompositionEngine.InstructionOutput

Description

Constructors

InstructionOutput()

Public Fields

InstructionOutputType *Type*

VariableDefinition *Definition*

VariableReference *Reference*

<<<

InstructionOutputType

PiRhoSoft.CompositionEngine.InstructionOutputType

Description

Values

InstructionOutputType *Ignore*

InstructionOutputType *Reference*

<<<

InstructionStore

PiRhoSoft.CompositionEngine.InstructionStore : [IVariableStore](#)

Description

Static Fields

string *ThisStoreName*

string *SceneStoreName*

string *InputStoreName*

string *OutputStoreName*

string *LocalStoreName*

Static Methods

```

bool IsInput(VariableReference variable)
bool IsOutput(VariableReference variable)
    # Constructors

InstructionStore(InstructionContext context, Object thisObject)
    # Public Properties

Object This (read only)
InstructionContext Context (read only)
VariableStore Inputs (read only)
VariableStore Outputs (read only)
VariableStore Locals (read only)
    # Public Methods

void SetContext(InstructionContext context)
void ChangeThis(Object thisObject)
void WriteInputs(IList<InstructionInput> inputs)
void WriteOutputs(IList<InstructionOutput> outputs)
void ReadOutputs(IList<InstructionOutput> outputs)
VariableValue GetVariable(string name) (virtual)
SetVariableResult SetVariable(string name, VariableValue value) (virtual)
IEnumerable<string> GetVariableNames() (virtual)
    <<<

```

IntegerVariableSource

PiRhoSoft.CompositionEngine.IntegerVariableSource : [VariableSource](#)<int>

Description

Constructors

```

IntegerVariableSource()
IntegerVariableSource(int defaultValue)
    # Protected Methods

bool TryGetValue(VariableValue variable, int value (out)) (virtual)
    <<<

```

Interface

PiRhoSoft.CompositionEngine.Interface : [MonoBehaviour](#)

Description

Constructors

Interface()

Public Fields

string *Name*

InterfaceControlDictionary *InterfaceControls*

Public Methods

void **Activate()**

void **Deactivate()**

ControlType **GetControl<ControlType>(string name)**

Protected Methods

void **Setup()** (*virtual*)

void **Teardown()** (*virtual*)

void **UpdateInput()** (*virtual*)

<<<

InterfaceBinding

PiRhoSoft.CompositionEngine.InterfaceBinding : [MonoBehaviour](#)

Description

Static Methods

void **UpdateBindings**([GameObject](#) *obj*, [IVariableStore](#) *variables*, **string** *group*, [BindingAnimationStatus](#) *status*)

Public Fields

string *BindingGroup*

Public Methods

void **UpdateBinding**([IVariableStore](#) *variables*, [BindingAnimationStatus](#) *status*) (*abstract*)

void **UpdateBinding**([IVariableStore](#) *store*, **string** *group*, [BindingAnimationStatus](#) *status*)

Protected Constructors

InterfaceBinding()

<<<

InterfaceControl

PiRhoSoft.CompositionEngine.InterfaceControl : [MonoBehaviour](#)

Description

Constructors

InterfaceControl()

Public Fields

[DependentObjectList](#) *DependentObjects*

Public Properties

[IVariableStore](#) *Variables (read only)*

bool *IsActive (read only)*

Public Methods

void **Activate()**

void **Deactivate()**

void **UpdateBindings**([IVariableStore](#) *variables*, **string** *group*, [BindingAnimationStatus](#) *status*)
(*virtual*)

Protected Methods

void **Setup()** (*virtual*)

void **Teardown()** (*virtual*)

<<<

InterfaceControlDictionary

PiRhoSoft.CompositionEngine.InterfaceControlDictionary : [SerializedDictionary](#)<string, string>

Description

Constructors

InterfaceControlDictionary()

<<<

InterfaceManager

PiRhoSoft.CompositionEngine.InterfaceManager : [SingletonBehaviour](#)<[InterfaceManager](#)>

Description

Constructors

InterfaceManager()

Public Fields

string *HorizontalAxis*

string *VerticalAxis*

string *AcceptButton*

string *CancelButton*

string *StartButton*

KeyCode *AcceptKey*

KeyCode *CancelKey*

KeyCode *StartKey*

Public Properties

Interface *FocusedInterface (read only)*

ButtonState *Left (read only)*

ButtonState *Right (read only)*

ButtonState *Up (read only)*

ButtonState *Down (read only)*

ButtonState *Accept (read only)*

ButtonState *Cancel (read only)*

ButtonState *Start (read only)*

Public Methods

InterfaceType **GetInterface<InterfaceType>(string name)**

Protected Methods

void **UpdateInput()** *(virtual)*

<<<

InterfaceReference

PiRhoSoft.CompositionEngine.InterfaceReference

Description

Constructors

InterfaceReference()

Public Fields

```

string InterfaceName
string ControlName
    # Public Methods

void Activate(Object context)
ControlType Activate<ControlType>(Object context)
void Deactivate(Object context)
InterfaceType GetInterface<InterfaceType>(Object context)
ControlType GetControl<ControlType>(Object context)
<<<

```

InvertOperator

PiRhoSoft.CompositionEngine.InvertOperator : [PrefixOperation](#)

Description

Constructors

```

InvertOperator()
    # Public Methods

VariableValue Evaluate(IVariableStore variables) (virtual)
<<<

```

ISequenceNode

PiRhoSoft.CompositionEngine.ISequenceNode

Description

ItemSelector

PiRhoSoft.CompositionEngine.ItemSelector : [MonoBehaviour](#), [IEventSystemHandler](#), [IPointerEnterHandler](#), [IPointerExitHandler](#), [IPointerDownHandler](#), [IPointerUpHandler](#)

Description

Constructors

ItemSelector()

Public Properties

[SelectionControl](#) *Selection (read only)*

int *Index (read only)*

Public Methods

void **OnPointerEnter**([PointerEventData](#) eventData) (virtual)

void **OnPointerExit**([PointerEventData](#) eventData) (virtual)

void **OnPointerDown**([PointerEventData](#) eventData) (virtual)

void **OnPointerUp**([PointerEventData](#) eventData) (virtual)

<<<

IterateNode

PiRhoSoft.CompositionEngine.IterateNode : [InstructionGraphNode](#), [ILoopNode](#)

Description

Constructors

IterateNode()

Public Fields

[InstructionGraphNode](#) *Loop*

Public Properties

[Color](#) *NodeColor (read only) (virtual)*

Protected Methods

[IEnumerator](#) **Run_**([InstructionGraph](#) graph, [InstructionStore](#) variables, **int** iteration) (virtual)

<<<

IVariableList

PiRhoSoft.CompositionEngine.IVariableList

Description

Public Properties

int *VariableCount* (read only) (abstract)

Public Methods

string *GetVariableName*(int *index*) (abstract)

VariableValue *GetVariableValue*(int *index*) (abstract)

SetVariableResult *SetVariableValue*(int *index*, **VariableValue** *value*) (abstract)

<<<

IVariableListener

PiRhoSoft.CompositionEngine.IVariableListener

Description

Public Methods

void *VariableChanged*(int *index*, **VariableValue** *value*) (abstract)

<<<

IVariableStore

PiRhoSoft.CompositionEngine.IVariableStore

Description

Public Methods

VariableValue *GetVariable*(string *name*) (abstract)

SetVariableResult *SetVariable*(string *name*, **VariableValue** *value*) (abstract)

IEnumerable<string> *GetVariableNames*() (abstract)

<<<

JoinEnumerator

PiRhoSoft.CompositionEngine.JoinEnumerator : **IEnumerator**

Description

Static Fields

int *MaximumIterations*

Constructors

JoinEnumerator(**IEnumerator** *coroutine*)

Public Properties

Object *Current (read only) (virtual)*

Public Methods

bool **MoveNext()** *(virtual)*

void **Reset()** *(virtual)*

<<<

LerpCommand

PiRhoSoft.CompositionEngine.LerpCommand : [Command](#)

Description

Constructors

LerpCommand()

Public Methods

VariableValue **Evaluate**(**IVariableStore** *variables*, **string** *name*, **List**<[Operation](#)> *parameters*) *(virtual)*

<<<

LessOperator

PiRhoSoft.CompositionEngine.LessOperator : [InfixOperation](#)

Description

Constructors

LessOperator()

Public Methods

VariableValue **Evaluate**(**IVariableStore** *variables*) *(virtual)*

<<<

LessOrEqualOperator

PiRhoSoft.CompositionEngine.LessOrEqualOperator : [InfixOperation](#)

Description

Constructors

LessOrEqualOperator()

Public Methods

VariableValue Evaluate(**IVariableStore** variables) (virtual)

<<<

LiteralOperation

PiRhoSoft.CompositionEngine.LiteralOperation : [Operation](#)

Description

Constructors

LiteralOperation(**VariableValue** value)

Public Methods

VariableValue Evaluate(**IVariableStore** variables) (virtual)

void ToString(**StringBuilder** builder) (virtual)

<<<

LocalVariableStore

PiRhoSoft.CompositionEngine.LocalVariableStore<T> : [VariableStore](#)

Description

Public Methods

VariableValue GetVariable(string name) (virtual)

SetVariableResult SetVariable(string name, **VariableValue** value) (virtual)

Protected Constructors

LocalVariableStore(T store)

Protected Fields

T_store

<<<

LogCommand

PiRhoSoft.CompositionEngine.LogCommand : [Command](#)

Description

Constructors

LogCommand()

Public Methods

VariableValue Evaluate(**IVariableStore** variables, **string** name, **List**<**Operation**> parameters) (virtual)

<<<

LookupOperation

PiRhoSoft.CompositionEngine.LookupOperation : [Operation](#)

Description

Constructors

LookupOperation(string variable)

Public Fields

VariableReference Reference

Public Methods

VariableValue Evaluate(**IVariableStore** variables) (virtual)

void ToString(StringBuilder builder) (virtual)

void GetInputs(List<**VariableDefinition**> inputs, **string** source) (virtual)

<<<

LoopNode

PiRhoSoft.CompositionEngine.LoopNode : [InstructionGraphNode](#), [ILoopNode](#)

Description

Constructors

LoopNode()

Public Fields

Expression *Condition*

InstructionGraphNode *Loop*

Public Properties

Color *NodeColor (read only) (virtual)*

Public Methods

void GetInputs(List<VariableDefinition> inputs) (virtual)

Protected Methods

IEnumerator Run_(InstructionGraph graph, InstructionStore variables, int iteration) (virtual)

<<<

MappedVariableStore

PiRhoSoft.CompositionEngine.MappedVariableStore : [IVariableStore](#), [IVariableList](#)

Description

Constructors

MappedVariableStore()

Public Properties

int VariableCount (read only) (virtual)

Public Methods

void Setup(VariableMap map, IVariableList[] lists)

VariableValue GetVariable(string name) (virtual)

SetVariableResult SetVariable(string name, VariableValue value) (virtual)

IEnumerable<string> GetVariableNames() (virtual)

string GetVariableName(int index) (virtual)

VariableValue GetVariableValue(int index) (virtual)

SetVariableResult SetVariableValue(int index, VariableValue value) (virtual)

<<<

MaxCommand

PiRhoSoft.CompositionEngine.MaxCommand : [Command](#)

Description

Constructors

MaxCommand()

Public Methods

VariableValue Evaluate(**IVariableStore** variables, string name, **List<Operation>** parameters)
(virtual)

<<<

Message

PiRhoSoft.CompositionEngine.Message

Description

Constructors

Message()

Public Fields

string Text

Public Properties

bool HasText (read only)

Public Methods

void GetInputs(**List<VariableDefinition>** inputs)

string GetText(**IVariableStore** variables)

<<<

MessageBinding

PiRhoSoft.CompositionEngine.MessageBinding : [InterfaceBinding](#)

Description

Constructors

MessageBinding()

Public Fields

Message Message

Public Properties

TextMeshProUGUI *Text (read only)*

Public Methods

void UpdateBinding(**IVariableStore** variables, **BindingAnimationStatus** status) (virtual)

<<<

MessageControl

PiRhoSoft.CompositionEngine.MessageControl : [InterfaceControl](#)

Description

Constructors

MessageControl()

Public Fields

TextMeshProUGUI *DisplayText*

[GameObject](#) *ContinueIndicator*

[GameObject](#) *FinishedIndicator*

Public Methods

[IEnumerator](#) **Show**(**IVariableStore** variables, **string** text, [MessageInteractionType](#) interaction, **bool** isLast)

Protected Methods

[IEnumerator](#) **Run**(**string** text, [MessageInteractionType](#) interaction, **bool** isLast) (virtual)

void Setup() (virtual)

void Teardown() (virtual)

void SetInteraction([MessageControlDisplay](#) display, [MessageInteractionType](#) interaction, **bool** isLast)

<<<

MessageControlDisplay

PiRhoSoft.CompositionEngine.MessageControlDisplay

Description

Values

[MessageControlDisplay](#) *None*

[MessageControlDisplay](#) *Continue*

[MessageControlDisplay](#) *Finished*

<<<

MessageInteractionType

PiRhoSoft.CompositionEngine.MessageInteractionType

Description

Values

MessageInteractionType *WaitForInput*

MessageInteractionType *WaitForFinalInput*

MessageInteractionType *WaitForDisplay*

MessageInteractionType *DontWait*

<<<

MessageNode

PiRhoSoft.CompositionEngine.MessageNode : [InstructionGraphNode](#)

Description

Constructors

MessageNode()

Public Fields

InstructionGraphNode *Next*

InterfaceReference *Control*

MessageInteractionType *Interaction*

bool *IsLast*

Message *Message*

Public Properties

Color *NodeColor (read only) (virtual)*

Public Methods

void **GetInputs**(**List**<**VariableDefinition**> *inputs*) *(virtual)*

Protected Methods

IEnumerator **Run_**(**InstructionGraph** *graph*, **InstructionStore** *variables*, **int** *iteration*) *(virtual)*

<<<

MinCommand

PiRhoSoft.CompositionEngine.MinCommand : [Command](#)

Description

Constructors

MinCommand()

Public Methods

VariableValue Evaluate(**IVariableStore** variables, **string** name, **List<Operation>** parameters)
(virtual)

<<<

MockupConnection

PiRhoSoft.CompositionEngine.MockupConnection

Description

Constructors

MockupConnection()

Public Fields

string Name

InstructionGraphNode Node

<<<

MockupConnectionList

PiRhoSoft.CompositionEngine.MockupConnectionList : [SerializedList<MockupConnection>](#)

Description

Constructors

MockupConnectionList()

<<<

MockupGraph

PiRhoSoft.CompositionEngine.MockupGraph : [InstructionGraph](#)

Description

Constructors

MockupGraph()

Public Fields

MockupConnectionList *EntryPoints*

Public Methods

void GetConnections(NodeData data) (virtual)

void SetConnection(ConnectionData connection, InstructionGraphNode target) (virtual)

Protected Methods

IEnumerator Run(InstructionStore variables) (virtual)

<<<

MockupNode

PiRhoSoft.CompositionEngine.MockupNode : [InstructionGraphNode](#)

Description

Constructors

MockupNode()

Public Fields

MockupConnectionList *Connections*

Color *DisplayColor*

Public Properties

Color *NodeColor (read only)* (virtual)

Public Methods

void GetConnections(NodeData data) (virtual)

void SetConnection(ConnectionData connection, InstructionGraphNode target) (virtual)

Protected Methods

IEnumerator Run_(InstructionGraph graph, InstructionStore variables, int iteration) (virtual)

<<<

ModuloAssignOperator

PiRhoSoft.CompositionEngine.ModuloAssignOperator : [AssignOperator](#)

Description

Constructors

ModuloAssignOperator()

Public Methods

VariableValue Evaluate(**IVariableStore** variables) (virtual)

<<<

ModuloOperator

PiRhoSoft.CompositionEngine.ModuloOperator : [InfixOperation](#)

Description

Static Methods

VariableValue Modulo(**VariableValue** left (ref), **VariableValue** right (ref))

Constructors

ModuloOperator()

Public Methods

VariableValue Evaluate(**IVariableStore** variables) (virtual)

<<<

MultiplyAssignOperator

PiRhoSoft.CompositionEngine.MultiplyAssignOperator : [AssignOperator](#)

Description

Constructors

MultiplyAssignOperator()

Public Methods

VariableValue Evaluate(**IVariableStore** variables) (virtual)

<<<

MultiplyOperator

PiRhoSoft.CompositionEngine.MultiplyOperator : [InfixOperation](#)

Description

Static Methods

VariableValue **Multiply**(**VariableValue** *left (ref)*, **VariableValue** *right (ref)*)

Constructors

MultiplyOperator()

Public Methods

VariableValue **Evaluate**(**IVariableStore** *variables*) (virtual)

<<<

NegateOperator

PiRhoSoft.CompositionEngine.NegateOperator : [PrefixOperation](#)

Description

Constructors

NegateOperator()

Public Methods

VariableValue **Evaluate**(**IVariableStore** *variables*) (virtual)

<<<

NodeData

PiRhoSoft.CompositionEngine.NodeData

Description

Static Fields

float *Width*

float *HeaderHeight*

float *LineHeight*

float *FooterHeight*

Constructors

NodeData(**InstructionGraphNode** *node*)

Public Fields

List<**ConnectionData**> *Connections*

Public Properties

InstructionGraphNode *Node (read only)*

Rect *Bounds (read only)*

Vector2 *Position*

float *InnerHeight*

Public Methods

void **ClearConnections()**

void **AddConnections**(**Object** *obj*)

void **AddConnection**(**string** *name*, **InstructionGraphNode** *to*)

void **AddConnection**(**string** *name*, **string** *key*, **InstructionGraphNode** *to*)

void **AddConnection**(**string** *name*, **int** *index*, **InstructionGraphNode** *to*)

<<<

NumberVariableSource

PiRhoSoft.CompositionEngine.NumberVariableSource : **VariableSource**<**float**>

Description

Constructors

NumberVariableSource()

NumberVariableSource(**float** *defaultValue*)

Protected Methods

bool **TryGetValue**(**VariableValue** *variable*, **float** *value (out)*) (*virtual*)

<<<

ObjectSource

PiRhoSoft.CompositionEngine.ObjectSource

Description

Values

ObjectSource *Scene*

ObjectSource *Asset*

<<<

ObjectVariableSource

PiRhoSoft.CompositionEngine.ObjectVariableSource<ObjectType> : VariableSource<ObjectType>

Description

Constructors

ObjectVariableSource()

Protected Methods

bool TryGetValue(VariableValue variable, ObjectType value (out)) (virtual)

<<<

Operation

PiRhoSoft.CompositionEngine.Operation

Description

Static Fields

string MismatchedMathType1Exception

string MismatchedMathType2Exception

string MismatchedBooleanType1Exception

string MismatchedBooleanType2Exception

Public Methods

VariableValue Evaluate(IVariableStore variables) (abstract)

void ToString(StringBuilder builder) (abstract)

void GetInputs(List<VariableDefinition> inputs, string source) (virtual)

void GetOutputs(List<VariableDefinition> outputs, string source) (virtual)

string ToString() (virtual)

Protected Constructors

Operation()

<<<

OperatorPrecedence

PiRhoSoft.CompositionEngine.OperatorPrecedence : ValueType

Description

Static Fields

OperatorPrecedence *Assignment*

OperatorPrecedence *Or*

OperatorPrecedence *And*

OperatorPrecedence *Ternary*

OperatorPrecedence *Equality*

OperatorPrecedence *Comparison*

OperatorPrecedence *Addition*

OperatorPrecedence *Multiplication*

OperatorPrecedence *Exponentiation*

Static Methods

OperatorPrecedence **LeftAssociative**(**int** value)

OperatorPrecedence **RightAssociative**(**int** value)

Public Properties

int Value (*read only*)

int AssociativeValue (*read only*)

<<<

OrAssignOperator

PiRhoSoft.CompositionEngine.OrAssignOperator : [AssignOperator](#)

Description

Constructors

OrAssignOperator()

Public Methods

VariableValue **Evaluate**(**IVariableStore** variables) (*virtual*)

<<<

OrOperator

PiRhoSoft.CompositionEngine.OrOperator : [InfixOperation](#)

Description

Static Methods

VariableValue Or(**VariableValue** left (ref), **VariableValue** right (ref))

Constructors

OrOperator()

Public Methods

VariableValue Evaluate(**IVariableStore** variables) (virtual)

<<<

ParameterStore

PiRhoSoft.CompositionEngine.ParameterStore : [VariableStore](#)

Description

Static Fields

string *ParameterName*

Constructors

ParameterStore()

Public Fields

List<**VariableValue**> *Parameters*

Public Methods

VariableValue GetVariable(string name) (virtual)

SetVariableResult SetVariable(string name, **VariableValue** value) (virtual)

IEnumerable<string> GetVariableNames() (virtual)

<<<

Pixelate

PiRhoSoft.CompositionEngine.Pixelate : [Transition](#)

Description

Constructors

Pixelate()

Public Fields

int *MaxAmount*

Public Methods

void Begin([TransitionPhase](#) phase) (virtual)

void Process(float time, [TransitionPhase](#) phase) (virtual)

Protected Methods

void Update() (virtual)

<<<

PlayAnimation

PiRhoSoft.CompositionEngine.PlayAnimation : [InstructionGraphNode](#)

Description

Constructors

PlayAnimation()

Public Fields

[InstructionGraphNode](#) Next

[AnimationClipVariableSource](#) Animation

bool WaitForCompletion

Public Properties

[Color](#) NodeColor (read only) (virtual)

Public Methods

void GetInputs([List](#)<[VariableDefinition](#)> inputs) (virtual)

Protected Methods

[IEnumerator](#) Run_([InstructionGraph](#) graph, [InstructionStore](#) variables, **int** iteration) (virtual)

<<<

PlayAnimationState

PiRhoSoft.CompositionEngine.PlayAnimationState : [InstructionGraphNode](#)

Description

Constructors

PlayAnimationState()

Public Fields

[InstructionGraphNode](#) *Next*

[StringVariableSource](#) *State*

Public Properties

[Color](#) *NodeColor (read only) (virtual)*

Public Methods

void **GetInputs**([List](#)<[VariableDefinition](#)> inputs) (virtual)

Protected Methods

IEnumerator **Run_**([InstructionGraph](#) graph, [InstructionStore](#) variables, **int** iteration) (virtual)

<<<

PlaybackState

PiRhoSoft.CompositionEngine.PlaybackState

Description

Values

[PlaybackState](#) *Running*

[PlaybackState](#) *Paused*

[PlaybackState](#) *Step*

[PlaybackState](#) *Stopped*

<<<

PlayTransition

PiRhoSoft.CompositionEngine.PlayTransition : [InstructionGraphNode](#)

Description

Constructors

PlayTransition()

Public Fields

[InstructionGraphNode](#) *Next*

[TransitionVariableSource](#) *Transition*

[TransitionPhase](#) *Phase*

bool *AutoFinish*

Public Properties

Color *NodeColor (read only) (virtual)*

Public Methods

void GetInputs(List<VariableDefinition> inputs) (virtual)

Protected Methods

IEnumerator Run_(InstructionGraph graph, InstructionStore variables, int iteration) (virtual)

<<<

PowCommand

PiRhoSoft.CompositionEngine.PowCommand : [Command](#)

Description

Constructors

PowCommand()

Public Methods

VariableValue Evaluate(IVariableStore variables, string name, List<Operation> parameters) (virtual)

<<<

PrefixOperation

PiRhoSoft.CompositionEngine.PrefixOperation : [Operation](#)

Description

Public Fields

string Symbol

Operation Right

Public Methods

void ToString(StringBuilder builder) (virtual)

void GetInputs(List<VariableDefinition> inputs, string source) (virtual)

Protected Constructors

PrefixOperation()

<<<

PrimaryAxis

PiRhoSoft.CompositionEngine.PrimaryAxis

Description

Values

PrimaryAxis *Column*

PrimaryAxis *Row*

<<<

Property

PiRhoSoft.CompositionEngine.Property<OwnerType>

Description

Constructors

Property()

Public Fields

string *Name*

Func<OwnerType, OwnerType> *Getter*

Func<OwnerType, OwnerType, OwnerType> *Setter*

<<<

PropertyList

PiRhoSoft.CompositionEngine.PropertyList<OwnerType> : [IVariableList](#)

Description

Constructors

PropertyList(OwnerType owner, [PropertyMap](#)<OwnerType> map)

Public Properties

int *VariableCount (read only) (virtual)*

Public Methods

string **GetVariableName(int index)** *(virtual)*

[VariableValue](#) **GetVariableValue(int index)** *(virtual)*

SetVariableResult SetVariableValue(int index, **VariableValue** value) (virtual)

<<<

PropertyMap

PiRhoSoft.CompositionEngine.PropertyMap

Description

Public Properties

int PropertyCount (read only) (abstract)

Public Methods

string GetPropertyName(int index) (abstract)

IVariableList CreateList(Object owner) (abstract)

Protected Constructors

PropertyMap()

<<<

PropertyMap

PiRhoSoft.CompositionEngine.PropertyMap<OwnerType> : **PropertyMap**

Description

Constructors

PropertyMap()

Public Fields

List<**Property**<OwnerType>> Properties

Public Properties

int PropertyCount (read only) (virtual)

Public Methods

string GetPropertyName(int index) (virtual)

IVariableList CreateList(Object owner) (virtual)

PropertyMap<OwnerType> Add(string name, Func<OwnerType, OwnerType> getter, Func<OwnerType, OwnerType, OwnerType> setter)

<<<

RandomCommand

PiRhoSoft.CompositionEngine.RandomCommand : [Command](#)

Description

Constructors

RandomCommand()

Public Methods

VariableValue Evaluate(**IVariableStore** variables, **string** name, **List<Operation>** parameters) (virtual)

<<<

ReadOnlyStore

PiRhoSoft.CompositionEngine.ReadOnlyStore : [VariableStore](#)

Description

Constructors

ReadOnlyStore()

Public Methods

SetVariableResult SetVariable(**string** name, **VariableValue** value) (virtual)

<<<

RoundCommand

PiRhoSoft.CompositionEngine.RoundCommand : [Command](#)

Description

Constructors

RoundCommand()

Public Methods

VariableValue Evaluate(**IVariableStore** variables, **string** name, **List<Operation>** parameters) (virtual)

<<<

SceneVariableStore

PiRhoSoft.CompositionEngine.SceneVariableStore : [IVariableStore](#)

Description

Constructors

SceneVariableStore()

Public Methods

[VariableValue](#) **GetVariable**(string name) (virtual)

[SetVariableResult](#) **SetVariable**(string name, [VariableValue](#) value) (virtual)

[IEnumerable](#)<string> **GetVariableNames**() (virtual)

<<<

ScopedGraph

PiRhoSoft.CompositionEngine.ScopedGraph : [InstructionGraph](#)

Description

Constructors

ScopedGraph()

Public Fields

[InstructionGraphNode](#) *Enter*

[InstructionGraphNode](#) *Process*

[InstructionGraphNode](#) *Exit*

Protected Methods

[IEnumerator](#) **Run**([InstructionStore](#) variables) (virtual)

<<<

ScrollSelection

PiRhoSoft.CompositionEngine.ScrollSelection : [SelectionControl](#)

Description

Constructors

ScrollSelection()

Public Fields

float *ScrollSpeed*

int *DisplayedVertical*

int *DisplayedHorizontal*

Public Methods

void **MoveFocusUp()** (*virtual*)

void **MoveFocusDown()** (*virtual*)

void **MoveFocusLeft()** (*virtual*)

void **MoveFocusRight()** (*virtual*)

Protected Methods

void **Setup()** (*virtual*)

Transform **GetItemParent()** (*virtual*)

<<<

SelectionBinding

PiRhoSoft.CompositionEngine.SelectionBinding : [MonoBehaviour](#)

Description

Constructors

SelectionBinding()

Public Fields

SelectionControl *SelectionControl*

string *BindingGroup*

<<<

SelectionControl

PiRhoSoft.CompositionEngine.SelectionControl : [InterfaceControl](#)

Description

Constructors

SelectionControl()

Public Fields

bool *VerticalWrapping*

bool *HorizontalWrapping*

PrimaryAxis *PrimaryAxis*

int *ColumnCount*

int *RowCount*

Public Properties

SelectedItem *FocusedItem (read only)*

IVariableStore *FocusedVariables (read only)*

SelectedItem *SelectedItem (read only)*

IVariableStore *SelectedVariables (read only)*

Public Methods

void **UpdateBindings**(**IVariableStore** *variables*, **string** *group*, **BindingAnimationStatus** *status*)
(*virtual*)

IEnumerator **MakeSelection**(**IVariableStore** *variables*, **IEnumerable**<**SelectedItem**> *items*, **bool** *isSelectionRequired*)

void **Close**()

void **SelectItem**(**int** *index*)

void **MoveFocus**(**int** *index*)

void **MoveFocusUp**() (*virtual*)

void **MoveFocusDown**() (*virtual*)

void **MoveFocusLeft**() (*virtual*)

void **MoveFocusRight**() (*virtual*)

bool **MoveFocusToStart**()

bool **MoveFocusToEnd**()

bool **MoveFocusToTop**()

bool **MoveFocusToBottom**()

bool **MoveFocusToLeft**()

bool **MoveFocusToRight**()

bool **MoveFocusToLocation**(**int** *column*, **int** *row*)

bool **SetFocusToValidLocation**(**int** *startingColumn*, **int** *startingRow*)

Protected Fields

int *_columnCount*

int *_rowCount*

int *_columnIndex*

int *_rowIndex*

List<**MenuItem**> *_items*

MenuItem *_focusedItem*

MenuItem *_selectedItem*

Protected Methods

```

void Teardown() (virtual)
Transform GetItemParent() (virtual)
MenuItem GetItem(int column, int row)
void FocusItem(MenuItem item) (virtual)
void BlurItem(MenuItem item) (virtual)
void MoveFocus(int change, bool wrap, int count, int depth, int index (ref), int column (ref), int
row (ref))
void SetFocus(int column, int row, bool force)
bool SetFocusToValidColumn(int startingColumn, int startingRow)
bool SetFocusToValidColumnInRow(int startingColumn, int row)
bool SetFocusToValidRow(int startingColumn, int startingRow)
bool SetFocusToValidRowInColumn(int column, int startingRow)
bool IsLocationFocusable(int column, int row)

<<<

```

SelectionItem

PiRhoSoft.CompositionEngine.SelectionItem : [IVariableStore](#)

Description

Constructors

SelectionItem()

Public Fields

[VariableReference](#) Item

[ObjectSource](#) Source

string Name

[GameObject](#) Template

string Label

bool Expand

Public Properties

string Id (read only)

Public Methods

[VariableValue](#) GetVariable(string name) (virtual)

[SetVariableResult](#) SetVariable(string name, [VariableValue](#) value) (virtual)

[IEnumerable](#)<string> GetVariableNames() (virtual)

<<<

SelectionNode

PiRhoSoft.CompositionEngine.SelectionNode : [InstructionGraphNode](#)

Description

Constructors

SelectionNode()

Public Fields

[InstructionGraphNode](#) *OnCanceled*

[InterfaceReference](#) *Control*

bool *IsSelectionRequired*

[SelectionNodeItemList](#) *Items*

Public Properties

[Color](#) *NodeColor (read only) (virtual)*

Public Methods

void **GetConnections**([NodeData](#) data) (virtual)

void **SetConnection**([ConnectionData](#) connection, [InstructionGraphNode](#) target) (virtual)

Protected Methods

[IEnumerator](#) **Run_**([InstructionGraph](#) graph, [InstructionStore](#) variables, **int** iteration) (virtual)

<<<

SelectionNodeItem

PiRhoSoft.CompositionEngine.SelectionNodeItem : [SelectionItem](#)

Description

Constructors

SelectionNodeItem()

Public Fields

[InstructionGraphNode](#) *OnSelected*

<<<

SelectionNodeItemList

PiRhoSoft.CompositionEngine.SelectionNodeItemList : [SerializedList](#)<[SelectionNodeItem](#)>

Description

Constructors

SelectionNodeItemList()

<<<

SequenceNode

PiRhoSoft.CompositionEngine.SequenceNode : [InstructionGraphNode](#), [ISequenceNode](#)

Description

Constructors

SequenceNode()

Public Fields

[InstructionGraphNodeList](#) *Sequence*

Public Properties

[Color](#) *NodeColor (read only) (virtual)*

Protected Methods

[IEnumerator](#) **Run_**([InstructionGraph](#) *graph*, [InstructionStore](#) *variables*, **int** *iteration*) (virtual)

<<<

SerializedVariable

PiRhoSoft.CompositionEngine.SerializedVariable

Description

Constructors

SerializedVariable()

Public Fields

string *Name*

[VariableType](#) *Type*

string *Data*

[Object](#) *Object*

Public Methods

```

void SetVariable(Variable variable)
void SetValue(VariableValue value)
Variable GetVariable()
VariableValue GetValue()

```

<<<

SetAnimationParameter

PiRhoSoft.CompositionEngine.SetAnimationParameter : [InstructionGraphNode](#)

Description

Constructors

SetAnimationParameter()

Public Fields

[InstructionGraphNode](#) *Next*

[StringVariableSource](#) *Parameter*

[AnimatorControllerParameterType](#) *Type*

[BooleanVariableSource](#) *BoolValue*

[IntegerVariableSource](#) *IntValue*

[NumberVariableSource](#) *FloatValue*

Public Properties

[Color](#) *NodeColor (read only) (virtual)*

Public Methods

void GetInputs(List<[VariableDefinition](#)> inputs) (virtual)

Protected Methods

[IEnumerator](#) Run_([InstructionGraph](#) graph, [InstructionStore](#) variables, **int iteration) (virtual)**

<<<

SetVariableResult

PiRhoSoft.CompositionEngine.SetVariableResult

Description

Values

SetVariableResult *Success*
SetVariableResult *NotFound*
SetVariableResult *ReadOnly*
SetVariableResult *TypeMismatch*

<<<

ShowControlNode

PiRhoSoft.CompositionEngine.ShowControlNode : [InstructionGraphNode](#)

Description

Constructors

ShowControlNode()

Public Fields

InstructionGraphNode *Next*

InterfaceReference *Control*

Public Properties

Color *NodeColor (read only) (virtual)*

Protected Methods

IEnumerator **Run_**(**InstructionGraph** graph, **InstructionStore** variables, **int** iteration) (virtual)

<<<

SignCommand

PiRhoSoft.CompositionEngine.SignCommand : [Command](#)

Description

Constructors

SignCommand()

Public Methods

VariableValue **Evaluate**(**IVariableStore** variables, **string** name, **List**<**Operation**> parameters) (virtual)

<<<

SimpleGraph

PiRhoSoft.CompositionEngine.SimpleGraph : [InstructionGraph](#)

Description

Constructors

SimpleGraph()

Public Fields

InstructionGraphNode *Process*

Protected Methods

IEnumerator **Run**(**InstructionStore** variables) (virtual)

<<<

SinCommand

PiRhoSoft.CompositionEngine.SinCommand : [Command](#)

Description

Constructors

SinCommand()

Public Methods

VariableValue **Evaluate**(**IVariableStore** variables, **string** name, **List**<**Operation**> parameters) (virtual)

<<<

SqrtCommand

PiRhoSoft.CompositionEngine.SqrtCommand : [Command](#)

Description

Constructors

SqrtCommand()

Public Methods

VariableValue **Evaluate**(**IVariableStore** variables, **string** name, **List**<**Operation**> parameters) (virtual)

<<<

StringVariableSource

PiRhoSoft.CompositionEngine.StringVariableSource : [VariableSource](#)<string>

Description

Constructors

StringVariableSource()

StringVariableSource(string defaultValue)

Protected Methods

bool TryGetValue(VariableValue variable, string value (out)) (virtual)

<<<

SubtractAssignOperator

PiRhoSoft.CompositionEngine.SubtractAssignOperator : [AssignOperator](#)

Description

Constructors

SubtractAssignOperator()

Public Methods

VariableValue Evaluate(IVariableStore variables) (virtual)

<<<

SubtractOperator

PiRhoSoft.CompositionEngine.SubtractOperator : [InfixOperation](#)

Description

Static Methods

VariableValue Subtract(VariableValue left (ref), VariableValue right (ref))

Constructors

SubtractOperator()

Public Methods

VariableValue Evaluate(IVariableStore variables) (virtual)

<<<

TanCommand

PiRhoSoft.CompositionEngine.TanCommand : [Command](#)

Description

Constructors

TanCommand()

Public Methods

VariableValue Evaluate(**IVariableStore** variables, string name, **List<Operation>** parameters) (virtual)

<<<

TernaryOperator

PiRhoSoft.CompositionEngine.TernaryOperator : [Operation](#)

Description

Constructors

TernaryOperator()

Public Fields

Operation Condition

Operation TrueBranch

Operation FalseBranch

Public Methods

void ToString(StringBuilder builder) (virtual)

VariableValue Evaluate(**IVariableStore** variables) (virtual)

<<<

TextBinding

PiRhoSoft.CompositionEngine.TextBinding : [InterfaceBinding](#)

Description

Constructors

TextBinding()

Public Fields

VariableReference *Variable*

Public Methods

void UpdateBinding(**IVariableStore** variables, **BindingAnimationStatus** status) (virtual)

<<<

Transition

PiRhoSoft.CompositionEngine.Transition : **ScriptableObject**

Description

Public Fields

float *Duration*

Public Methods

void Begin(**TransitionPhase** phase) (virtual)

void Process(**float** time, **TransitionPhase** phase) (virtual)

void End() (virtual)

void Render(**TransitionRenderer** renderer, **RenderTexture** source, **RenderTexture** destination) (virtual)

Protected Constructors

Transition()

Protected Properties

Material *Material* (read only)

Protected Methods

void SetShader(**string** name)

void Update() (virtual)

<<<

TransitionList

PiRhoSoft.CompositionEngine.TransitionList : **SerializedList**<**Transition**>

Description

Constructors

TransitionList()

<<<

TransitionManager

PiRhoSoft.CompositionEngine.TransitionManager : [GlobalBehaviour](#)<[TransitionManager](#)>

Description

Constructors

TransitionManager()

Public Properties

[Transition](#) *CurrentTransition (read only)*

[TransitionRenderer](#) *CurrentRenderer (read only)*

Public Methods

void AddRenderer([TransitionRenderer](#) renderer)

void RemoveRenderer([TransitionRenderer](#) renderer)

IEnumerator RunTransition([Transition](#) transition, [TransitionPhase](#) phase)

IEnumerator StartTransition([Transition](#) transition, [TransitionPhase](#) phase)

void EndTransition()

<<<

TransitionPhase

PiRhoSoft.CompositionEngine.TransitionPhase

Description

Values

[TransitionPhase](#) *Out*

[TransitionPhase](#) *Obscure*

[TransitionPhase](#) *In*

<<<

TransitionRenderer

PiRhoSoft.CompositionEngine.TransitionRenderer : [MonoBehaviour](#)

Description

Constructors

TransitionRenderer()

<<<

TransitionVariableSource

PiRhoSoft.CompositionEngine.TransitionVariableSource : [ObjectVariableSource](#)<[Transition](#)>

Description

Constructors

TransitionVariableSource()

<<<

TruncateCommand

PiRhoSoft.CompositionEngine.TruncateCommand : [Command](#)

Description

Constructors

TruncateCommand()

Public Methods

[VariableValue](#) **Evaluate**([IVariableStore](#) variables, **string** name, [List](#)<[Operation](#)> parameters)
(virtual)

<<<

UpdateBindingNode

PiRhoSoft.CompositionEngine.UpdateBindingNode : [InstructionGraphNode](#)

Description

Constructors

UpdateBindingNode()

Public Fields

[InstructionGraphNode](#) *Next*

[InterfaceReference](#) *Control*

string *Group*

bool *WaitForCompletion*

Public Properties

Color *NodeColor (read only) (virtual)*

Protected Methods

IEnumerator **Run_**(**InstructionGraph** graph, **InstructionStore** variables, **int** iteration) (virtual)

<<<

Variable

PiRhoSoft.CompositionEngine.Variable : ValueType

Description

Static Properties

Variable *Empty (read only)*

Static Methods

Variable **Create**(string name, **VariableValue** value)

Public Properties

string *Name (read only)*

VariableValue *Value (read only)*

<<<

VariableAvailabilitiesAttribute

PiRhoSoft.CompositionEngine.VariableAvailabilitiesAttribute : Attribute

Description

Constructors

VariableAvailabilitiesAttribute(string[] availabilities)

Public Fields

string[] *Availabilities*

<<<

VariableDefinition

PiRhoSoft.CompositionEngine.VariableDefinition : ValueType

Description

Static Fields

string *NotSaved*

string *Saved*

Static Methods

VariableDefinition Create(string name, **VariableType** type, string availability, **Expression** initializer)

VariableDefinition Create(string name, int minimum, int maximum, string availability, **Expression** initializer)

VariableDefinition Create(string name, float minimum, float maximum, string availability, **Expression** initializer)

VariableDefinition Create(string name, string values, string availability, **Expression** initializer)

VariableDefinition Create<T>(string name, string availability, **Expression** initializer)

VariableDefinition Create(string name, **VariableType** type, bool constrainRange, float minimum, float maximum, string typeConstraint, string availability, **Expression** initializer)

Public Properties

string *Name (read only)*

VariableType *Type (read only)*

string *Availability (read only)*

Expression *Initializer (read only)*

bool *UseRangeConstraint (read only)*

float *MinimumConstraint (read only)*

float *MaximumConstraint (read only)*

string *TypeConstraint (read only)*

Public Methods

Variable Generate(**IVariableStore** variables)

<<<

VariableInitializerAttribute

PiRhoSoft.CompositionEngine.VariableInitializerAttribute : Attribute

Description

Constructors

VariableInitializerAttribute(**VariableInitializerType** type)

Public Fields

VariableInitializerType *Type*

<<<

VariableInitializerType

PiRhoSoft.CompositionEngine.VariableInitializerType

Description

Values

VariableInitializerType *Expression*

VariableInitializerType *DefaultValue*

VariableInitializerType *None*

<<<

VariableList

PiRhoSoft.CompositionEngine.VariableList : [IVariableList](#)

Description

Constructors

VariableList()

Public Properties

VariableSchema *Schema (read only)*

IVariableStore *Owner (read only)*

bool *NeedsUpdate (read only)*

int *VariableCount (read only) (virtual)*

Public Methods

void LoadFrom([VariableList](#) variables, **string** availability)

void SaveTo([VariableList](#) variables, **string** availability)

void Setup([VariableSchema](#) schema, [IVariableStore](#) owner)

void Update()

void Reset(int index)

void Reset(**string** availability)

void Reset([IList](#)<**string**> variables)

void Clear()

string **GetVariableName**(int index) (virtual)

VariableValue **GetVariableValue**(int index) (virtual)

SetVariableResult SetVariableValue(int index, **VariableValue** value) (virtual)

<<<

VariableListener

PiRhoSoft.CompositionEngine.VariableListener : **IVariableList**

Description

Constructors

VariableListener(**IVariableListener** listener, **IVariableList** list)

Public Properties

int VariableCount (read only) (virtual)

Public Methods

string GetVariableName(int index) (virtual)

VariableValue GetVariableValue(int index) (virtual)

SetVariableResult SetVariableValue(int index, **VariableValue** value) (virtual)

<<<

VariableMap

PiRhoSoft.CompositionEngine.VariableMap

Description

Constructors

VariableMap(int version)

Public Properties

int Version (read only)

int Count (read only)

Public Methods

bool Contains(string name)

int GetIndex(string name)

bool TryGetIndex(string name, int index (out))

IEnumerable<string> GetNames()

VariableMap Add(**VariableSchema** schema)

VariableMap Add(**PropertyMap** map)

<<<

VariableReference

PiRhoSoft.CompositionEngine.VariableReference

Description

Constructors

VariableReference()

VariableReference(string variable)

Public Properties

bool *IsAssigned (read only)*

string *StoreName (read only)*

string *RootName (read only)*

Public Methods

void **Update(string variable)**

VariableValue **GetValue(IVariableStore variables)**

SetVariableResult **SetValue(IVariableStore variables, VariableValue value)**

string **ToString()** *(virtual)*

<<<

VariableSchema

PiRhoSoft.CompositionEngine.VariableSchema

Description

Constructors

VariableSchema()

Public Properties

int *Version (read only)*

int *Count (read only)*

VariableDefinition *Item*

Public Methods

int **GetIndex(string name)**

bool **HasDefinition(string name)**

bool **AddDefinition(string name, VariableType type)**

void **RemoveDefinition(int index)**

<<<

VariableSource

PiRhoSoft.CompositionEngine.VariableSource<T>

Description

Public Fields

[VariableSourceType](#) *Type*

[VariableReference](#) *Reference*

T *Value*

Public Methods

void GetInputs([List](#)<[VariableDefinition](#)> inputs)

bool TryGetValue([IVariableStore](#) variables, [Object](#) context, **T** value (out))

Protected Constructors

VariableSource()

Protected Methods

bool TryGetValue([VariableValue](#) variable, **T** Value (out)) (abstract)

<<<

VariableSourceType

PiRhoSoft.CompositionEngine.VariableSourceType

Description

Values

[VariableSourceType](#) *Value*

[VariableSourceType](#) *Reference*

<<<

VariableStore

PiRhoSoft.CompositionEngine.VariableStore : [IVariableStore](#)

Description

Constructors

VariableStore()

Public Properties

List<**Variable**> *Variables (read only)*

Dictionary<**string**, **string**> *Map (read only)*

Public Methods

void **AddVariable**(**string** name, **VariableValue** value)

VariableValue **GetVariable**(**string** name) *(virtual)*

SetVariableResult **SetVariable**(**string** name, **VariableValue** value) *(virtual)*

IEnumerable<**string**> **GetVariableNames**() *(virtual)*

void **Clear**()

Protected Methods

SetVariableResult **SetVariable**(**string** name, **VariableValue** value, **bool** allowAdd)

<<<

VariableType

PiRhoSoft.CompositionEngine.VariableType

Description

Values

VariableType *Empty*

VariableType *Boolean*

VariableType *Integer*

VariableType *Number*

VariableType *String*

VariableType *Object*

VariableType *Store*

VariableType *Null*

<<<

VariableValue

PiRhoSoft.CompositionEngine.VariableValue : ValueType, IComparable<string>, IComparable<float>, IComparable<int>, IComparable<bool>, IComparable<VariableValue>, IEquatable<string>, IEquatable<float>, IEquatable<int>, IEquatable<bool>, IEquatable<VariableValue>

Description

Static Fields

VariableValue *Empty*

Static Methods

VariableType **GetType**(Type type)

VariableValue **Create**(VariableType type)

VariableValue **Create**<T>(T value)

Public Properties

VariableType Type (read only)

bool Boolean (read only)

int Integer (read only)

float Number (read only)

string String (read only)

Object RawObject (read only)

Object Object (read only)

IVariableStore Store (read only)

Public Methods

string ToString() (virtual)

bool TryGetBoolean(bool value (out))

bool TryGetInteger(int value (out))

bool TryGetNumber(float value (out))

bool TryGetString(string value (out))

bool TryGetObject(Object value (out))

bool TryGetObject<T>(T obj (out))

bool TryGetStore(IVariableStore variables (out))

bool Equals(VariableValue other) (virtual)

bool Equals(bool value) (virtual)

bool Equals(int value) (virtual)

bool Equals(float value) (virtual)

bool Equals(string value) (virtual)

bool Equals(Object value)

bool Equals(IVariableStore value)

bool Equals(Object other) (virtual)

int CompareTo(VariableValue other) (virtual)

int CompareTo(bool value) (virtual)

int CompareTo(int value) (virtual)

int CompareTo(float value) (virtual)

int CompareTo(string value) (virtual)

int CompareTo(Object unityObject)

```

int CompareTo(IVariableStore store)
int GetHashCode() (virtual)
string Write()
void Read(string value)
<<<

```

VisibleBinding

PiRhoSoft.CompositionEngine.VisibleBinding : [InterfaceBinding](#)

Description

Constructors

```

VisibleBinding()
# Public Fields

```

```

Expression Condition
# Public Methods

```

```

void UpdateBinding(IVariableStore variables, BindingAnimationStatus status) (virtual)
<<<

```

WaitNode

PiRhoSoft.CompositionEngine.WaitNode : [InstructionGraphNode](#)

Description

Constructors

```

WaitNode()
# Public Fields

```

```

InstructionGraphNode Next
float Time
# Public Properties

```

```

Color NodeColor (read only) (virtual)
# Protected Methods

```

```

IEnumerator Run_(InstructionGraph graph, InstructionStore variables, int iteration) (virtual)
<<<

```

WritableStore

PiRhoSoft.CompositionEngine.WritableStore : [VariableStore](#)

Description

Constructors

WritableStore()

Public Methods

[SetVariableResult](#) **SetVariable**(string *name*, [VariableValue](#) *value*) (virtual)