

Unity Composition Reference

PiRho Soft

AbsCommand	1
Description	1
Constructors	1
AcosCommand	1
Description	1
Constructors	1
AddAssignOperator	1
Description	1
Constructors	1
AddOperator	2
Description	2
Static Methods	2
AndAssignOperator	2
Description	2
Constructors	2
AndOperator	2
Description	2
Static Methods	2
AsinCommand	3
Description	3
Constructors	3
AssignOperator	3
Description	3
Constructors	3
AtanCommand	3
Description	3
Constructors	4
AxisInput	4
Description	4
Constructors	4
BarBinding	4
Description	4
Constructors	4
BindingControl	5
Description	5
Constructors	5
BindingUpdater	5
Description	5
Constructors	5
BooleanVariableSource	5
Description	5

Constructors	5
BranchNode	6
Description	6
Constructors	6
BreakNode	6
Description	6
Constructors	6
ButtonInput	7
Description	7
Constructors	7
CeilingCommand	7
Description	7
Constructors	7
ClampCommand	7
Description	8
Constructors	8
Command	8
Description	8
Static Fields	8
CommandEvaluationException	8
Description	8
Constructors	8
CommandOperation	9
Description	9
Constructors	9
CommandSet	9
Description	9
Constructors	9
Composition	10
Description	10
Static Fields	10
CompositionManager	10
Description	10
Static Fields	10
ConditionalNode	10
Description	11
Constructors	11
ConnectionData	11
Description	11
Constructors	11
ConstantCommand	12

Description	12
Constructors	12
CosCommand	12
Description	12
Constructors	12
CreateInstructionGraphNodeMenuAttribute	12
Description	12
Constructors	13
CreateObjectNode	13
Description	13
Constructors	13
CustomCommand	13
Description	13
Constructors	14
CustomCommandList	14
Description	14
Constructors	14
DependentObjectList	14
Description	14
Constructors	14
DestroyObjectNode	14
Description	14
Constructors	14
DisableObjectNode	15
Description	15
Constructors	15
DivideAssignOperator	15
Description	15
Constructors	16
DivideOperator	16
Description	16
Static Methods	16
EnableObjectNode	16
Description	16
Constructors	16
EqualOperator	17
Description	17
Constructors	17
ExponentAssignOperator	17
Description	17
Constructors	17

ExponentOperator	17
Description	17
Static Methods	17
Expression	18
Description	18
Constructors	18
ExpressionCommand	18
Description	18
Static Fields	18
ExpressionCompilationResult	19
Description	19
Public Fields	19
ExpressionEvaluationException	19
Description	19
Constructors	19
ExpressionLexer	19
Description	19
Static Methods	19
ExpressionNode	20
Description	20
Constructors	20
ExpressionParseException	20
Description	20
Constructors	20
ExpressionParser	20
Description	21
Static Methods	21
ExpressionToken	21
Description	21
Constructors	21
ExpressionTokenizeException	21
Description	21
Constructors	21
ExpressionTokenType	22
Description	22
Values	22
FloorCommand	22
Description	22
Constructors	22
FocusIndicator	22
Description	23

Constructors	23
GreaterOperator	23
Description	23
Constructors	23
GreaterOrEqualOperator	23
Description	23
Constructors	23
HideControlNode	23
Description	23
Constructors	24
IndexedVariableStore	24
Description	24
Public Properties	24
ImageBinding	24
Description	24
Static Fields	24
IndexedVariableStore	25
Description	25
Static Fields	25
IndexedVariableStore	25
Description	25
Constructors	25
InequalOperator	25
Description	26
Constructors	26
InfixOperation	26
Description	26
Public Fields	26
Instruction	26
Description	26
Public Properties	26
InstructionCaller	27
Description	27
Constructors	27
InstructionContext	27
Description	27
Constructors	27
InstructionDictionary	28
Description	28
Constructors	28
InstructionGraph	28

Description	28
Static Fields	28
InstructionGraphExecutionMode	29
Description	29
Values	29
InstructionGraphNode	29
Description	29
Public Fields	29
InstructionGraphNodeDictionary	30
Description	30
Constructors	30
InstructionGraphNodeList	30
Description	30
Constructors	30
InstructionInput	30
Description	30
Constructors	30
InstructionInputType	30
Description	31
Values	31
InstructionList	31
Description	31
Constructors	31
InstructionNode	31
Description	31
Constructors	31
InstructionOutput	32
Description	32
Constructors	32
InstructionOutputType	32
Description	32
Values	32
InstructionStore	32
Description	32
Static Fields	32
IntegerVariableSource	33
Description	33
Constructors	33
Interface	33
Description	33
Constructors	33

InterfaceBinding	34
Description	34
Static Methods	34
InterfaceControl	34
Description	34
Constructors	34
InterfaceControlDictionary	35
Description	35
Constructors	35
InterfaceManager	35
Description	35
Constructors	35
InterfaceReference	36
Description	36
Constructors	36
InvertOperator	37
Description	37
Constructors	37
ItemSelector	37
Description	37
Constructors	37
IterateNode	37
Description	37
Constructors	38
IVariableList	38
Description	38
Public Properties	38
IVariableListener	38
Description	38
Public Methods	38
IVariableStore	39
Description	39
Public Methods	39
LerpCommand	39
Description	39
Constructors	39
LessOperator	39
Description	39
Constructors	39
LessOrEqualOperator	39
Description	40

Constructors	40
LiteralOperation	40
Description	40
Constructors	40
LocalVariableStore	40
Description	40
Public Methods	40
LogCommand	41
Description	41
Constructors	41
LookupOperation	41
Description	41
Constructors	41
LoopNode	41
Description	41
Constructors	41
MappedVariableStore	42
Description	42
Constructors	42
MaxCommand	42
Description	42
Constructors	43
Message	43
Description	43
Constructors	43
MessageBinding	43
Description	43
Constructors	43
MessageControl	44
Description	44
Constructors	44
MessageControlDisplay	44
Description	44
Values	44
MessageInteractionType	45
Description	45
Values	45
MessageNode	45
Description	45
Constructors	45
MinCommand	46

Description	46
Constructors	46
MockupConnection	46
Description	46
Constructors	46
MockupConnectionList	46
Description	46
Constructors	46
MockupGraph	46
Description	47
Constructors	47
MockupNode	47
Description	47
Constructors	47
ModuloAssignOperator	48
Description	48
Constructors	48
ModuloOperator	48
Description	48
Static Methods	48
MultiplyAssignOperator	48
Description	48
Constructors	48
MultiplyOperator	49
Description	49
Static Methods	49
NegateOperator	49
Description	49
Constructors	49
NodeData	49
Description	49
Static Fields	49
NumberVariableSource	50
Description	50
Constructors	50
ObjectSource	50
Description	50
Values	50
ObjectVariableSource	51
Description	51
Constructors	51

Operation	51
Description	51
Static Fields	51
OperatorPrecedence	51
Description	52
Static Fields	52
OrAssignOperator	52
Description	52
Constructors	52
OrOperator	52
Description	52
Static Methods	53
ParameterStore	53
Description	53
Static Fields	53
PlayAnimation	53
Description	53
Constructors	53
PlayAnimationState	54
Description	54
Constructors	54
PowCommand	55
Description	55
Constructors	55
PrefixOperation	55
Description	55
Public Fields	55
Property	55
Description	55
Constructors	56
PropertyList	56
Description	56
Constructors	56
PropertyMap	56
Description	56
Public Properties	56
PropertyMap	57
Description	57
Constructors	57
RandomCommand	57
Description	57

Constructors	57
RoundCommand	57
Description	58
Constructors	58
SceneVariableStore	58
Description	58
Constructors	58
ScopedGraph	58
Description	58
Constructors	58
ScrollSelection	59
Description	59
Constructors	59
SelectionBinding	59
Description	59
Constructors	59
SelectionControl	59
Description	60
Constructors	60
SelectionItem	61
Description	61
Constructors	61
SelectionNode	62
Description	62
Constructors	62
SelectionNodeItem	62
Description	62
Constructors	62
SelectionNodeItemList	63
Description	63
Constructors	63
SequenceNode	63
Description	63
Constructors	63
SerializedVariable	63
Description	63
Constructors	63
SetAnimationParameter	64
Description	64
Constructors	64
SetVariableResult	65

Description	65
Values	65
ShowControlNode	65
Description	65
Constructors	65
SignCommand	65
Description	65
Constructors	66
SimpleGraph	66
Description	66
Constructors	66
SinCommand	66
Description	66
Constructors	66
SqrtCommand	66
Description	67
Constructors	67
StringVariableSource	67
Description	67
Constructors	67
SubtractAssignOperator	67
Description	67
Constructors	67
SubtractOperator	67
Description	68
Static Methods	68
TanCommand	68
Description	68
Constructors	68
TernaryOperator	68
Description	68
Constructors	68
TextBinding	69
Description	69
Constructors	69
TruncateCommand	69
Description	69
Constructors	69
UpdateBindingNode	69
Description	69
Constructors	69

Variable	70
Description	70
Static Properties	70
VariableAvailabilitiesAttribute	70
Description	70
Constructors	70
VariableDefinition	71
Description	71
Static Fields	71
VariableInitializerAttribute	71
Description	72
Constructors	72
VariableInitializerType	72
Description	72
Values	72
VariableList	72
Description	72
Constructors	72
VariableListener	73
Description	73
Constructors	73
VariableMap	73
Description	73
Constructors	73
VariableReference	74
Description	74
Constructors	74
VariableSchema	74
Description	74
Constructors	74
VariableSource	75
Description	75
Public Fields	75
VariableSourceType	75
Description	75
Values	75
VariableStore	76
Description	76
Constructors	76
VariableType	76
Description	76

Values	76
VariableValue	76
Description	77
Static Fields	77
VisibleBinding	78
Description	78
Constructors	78
WaitNode	78
Description	78
Constructors	78

AbsCommand

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

Description

Constructors

AbsCommand()

Public Methods

Variable Value Evaluate(**I Variable Store** variables, string name, [List<Operation>](#) parameters)
(virtual)

<<<

AcosCommand

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

Description

Constructors

AcosCommand()

Public Methods

Variable Value Evaluate(**I Variable Store** variables, string name, [List<Operation>](#) parameters)
(virtual)

<<<

AddAssignOperator

PiRhoSoft.CompositionEngine.AddAssignOperator : [Assign Operator](#)

Description

Constructors

AddAssignOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (virtual)

<<<

AddOperator

PiRhoSoft.CompositionEngine.AddOperator : [Infix Operation](#)

Description

Static Methods

Variable Value Add(**Operation** expression, **Variable Value** left (*ref*), **Variable Value** right (*ref*))

Constructors

AddOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (*virtual*)

<<<

AndAssignOperator

PiRhoSoft.CompositionEngine.AndAssignOperator : [Assign Operator](#)

Description

Constructors

AndAssignOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (*virtual*)

<<<

AndOperator

PiRhoSoft.CompositionEngine.AndOperator : [Infix Operation](#)

Description

Static Methods

Variable Value And(**Operation** expression, **Variable Value** left (*ref*), **Variable Value** right (*ref*))

Constructors

AndOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (virtual)

<<<

AsinCommand

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

Description

Constructors

AsinCommand()

Public Methods

Variable Value Evaluate(**I Variable Store** variables, string name, [List<Operation>](#) parameters) (virtual)

<<<

AssignOperator

PiRhoSoft.CompositionEngine.AssignOperator : [Infix Operation](#)

Description

Constructors

AssignOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (virtual)

void GetInputs([List<Variable Definition>](#) inputs, string source) (virtual)

void GetOutputs([List<Variable Definition>](#) outputs, string source) (virtual)

Protected Methods

Variable Value Assign(**I Variable Store** variables, **Variable Value** value (ref))

<<<

AtanCommand

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

Description

Constructors

AtanCommand()

Public Methods

Variable Value Evaluate(**I Variable Store** variables, string name, **List<Operation>** parameters) (virtual)

<<<

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 : **Interface** **Binding**

Description

Constructors

BarBinding()

Public Fields

Variable Reference *AmountVariable*

Variable Reference *TotalVariable*

Gradient *FillColors*

Public Methods

void UpdateBinding(I Variable Store variables) (*virtual*)

<<<

BindingControl

PiRhoSoft.CompositionEngine.BindingControl : [Interface Control](#)

Description

Constructors

BindingControl()

Public Fields

Variable Reference *Binding*

Public Methods

void UpdateBindings(I Variable Store variables, string group) (*virtual*)

<<<

BindingUpdater

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

Description

Constructors

BindingUpdater()

Public Fields

string *Group*

<<<

BooleanVariableSource

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

Description

Constructors

BooleanVariableSource()

BooleanVariableSource(bool defaultValue)

Protected Methods

bool TryGetValue(**Variable Value** variable, bool value (out)) (virtual)

<<<

BranchNode

PiRhoSoft.CompositionEngine.BranchNode : [Instruction Graph Node](#)

Description

Constructors

BranchNode()

Public Fields

Expression Switch

Instruction Graph Node Dictionary Outputs

Instruction Graph Node Default

Public Properties

bool IsExecutionImmediate (read only) (virtual)

Instruction Graph Execution Mode ExecutionMode (read only) (virtual)

Public Methods

void GetInputs(**List**<**Variable Definition**> inputs) (virtual)

Protected Methods

IEnumerator Run_(**Instruction Graph** graph, **Instruction Store** variables, int iteration)
(virtual)

<<<

BreakNode

PiRhoSoft.CompositionEngine.BreakNode : [Instruction Graph Node](#)

Description

Constructors

BreakNode()

Public Fields

Instruction Graph Node Next

Public Properties

bool *IsExecutionImmediate* (read only) (virtual)

Instruction Graph Execution Mode *ExecutionMode* (read only) (virtual)

Protected Methods

IEnumerator *Run_*(**Instruction Graph** graph, **Instruction Store** 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

Variable Value *Evaluate*(**I Variable Store** variables, string name, [List<Operation>](#) parameters)
(virtual)

<<<

ClampCommand

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

Description

Constructors

ClampCommand()

Public Methods

Variable Value Evaluate(**I Variable Store** variables, string name, **List<Operation>** parameters)
(virtual)

<<<

Command

PiRhoSoft.CompositionEngine.Command

Description

Static Fields

string *WrongParameterCountException*

string *WrongParameterRangeException*

string *TooFewParametersException*

string *TooManyParametersException*

string *WrongParameterType1Exception*

string *WrongParameterType2Exception*

string *InvalidRangeException*

Public Methods

Variable Value Evaluate(**I Variable Store** 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<Variable Definition>](#) inputs, string source) (virtual)

[Variable Value](#) **Evaluate**([I Variable Store](#) variables) (virtual)

<<<

CommandSet

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

Description

Constructors

CommandSet()

Public Fields

[Custom Command List](#) *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*)

<<<

Composition

PiRhoSoft.CompositionEngine.Composition

Description

Static Fields

string *DocumentationUrl*

<<<

CompositionManager

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

Description

Static Fields

string *_processFailedError*

Constructors

CompositionManager()

Public Fields

Command Set *Commands*

Public Methods

void RunInstruction([Instruction](#) instruction, [Instruction Context](#) context, [I Variable Store](#) thisStore)

void RunInstruction([Instruction Caller](#) caller, [Instruction Context](#) context, [I Variable Store](#) thisStore)

<<<

ConditionalNode

PiRhoSoft.CompositionEngine.ConditionalNode : [Instruction Graph Node](#)

Description

Constructors

ConditionalNode()

Public Fields

Instruction Graph Node *OnTrue*

Instruction Graph Node *OnFalse*

Expression *Condition*

Public Properties

bool *IsExecutionImmediate (read only) (virtual)*

Instruction Graph Execution Mode *ExecutionMode (read only) (virtual)*

Public Methods

void **GetInputs**(**List**<**Variable Definition**> **inputs**) *(virtual)*

Protected Methods

IEnumerator **Run_**(**Instruction Graph** **graph**, **Instruction Store** **variables**, **int** **iteration**)
(virtual)

<<<

ConnectionData

PiRhoSoft.CompositionEngine.ConnectionData

Description

Constructors

ConnectionData(**string** **field**, **string** **key**, **int** **index**, **Instruction Graph Node** **from**, **Instruction Graph Node** **to**)

Public Properties

string *Field (read only)*

string *Key (read only)*

int *Index (read only)*

Instruction Graph Node *From (read only)*

Instruction Graph Node *To (read only)*

Node Data *Target (read only)*

string *Name (read only)*

Public Methods

```
void SetTarget(Node Data target)
void ChangeTarget(Node Data target)
void ApplyConnection(Object obj, Instruction Graph Node target)
<<<
```

ConstantCommand

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

Description

Constructors

ConstantCommand([Variable Value](#) value)

Public Properties

[Variable Value](#) Value (read only)

Public Methods

[Variable Value](#) Evaluate(I [Variable Store](#) variables, string name, [List<Operation>](#) parameters)
(virtual)

<<<

CosCommand

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

Description

Constructors

CosCommand()

Public Methods

[Variable Value](#) Evaluate(I [Variable Store](#) 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 : [Instruction Graph Node](#)

Description

Constructors

CreateObjectNode()

Public Fields

[Instruction Graph Node](#) *Next*

[GameObject](#) *Prefab*

string *ObjectName*

[Vector2](#) *Position*

[Variable Reference](#) *Parent*

Public Properties

bool *IsExecutionImmediate* (read only) (virtual)

[Instruction Graph Execution Mode](#) *ExecutionMode* (read only) (virtual)

Public Methods

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

Protected Methods

[IEnumerator](#) **Run_**([Instruction Graph](#) graph, [Instruction Store](#) variables, int iteration)
(virtual)

<<<

CustomCommand

PiRhoSoft.CompositionEngine.CustomCommand

Description

Constructors

CustomCommand()

Public Fields

string *Name*

Expression *Expression*

<<<

CustomCommandList

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

Description

Constructors

CustomCommandList()

<<<

DependentObjectList

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

Description

Constructors

DependentObjectList()

<<<

DestroyObjectNode

PiRhoSoft.CompositionEngine.DestroyObjectNode : [Instruction Graph Node](#)

Description

Constructors

DestroyObjectNode()

Public Fields

[Instruction Graph Node](#) *Next*

[Variable Reference](#) *Target*

Public Properties

bool *IsExecutionImmediate* (read only) (virtual)

Instruction Graph Execution Mode *ExecutionMode* (read only) (virtual)

Public Methods

void *GetInputs*(**List**<**Variable Definition**> inputs) (virtual)

Protected Methods

IEnumerator *Run_*(**Instruction Graph** graph, **Instruction Store** variables, int iteration)
(virtual)

<<<

DisableObjectNode

PiRhoSoft.CompositionEngine.DisableObjectNode : [Instruction Graph Node](#)

Description

Constructors

DisableObjectNode()

Public Fields

Instruction Graph Node *Next*

Variable Reference *Target*

Public Properties

bool *IsExecutionImmediate* (read only) (virtual)

Instruction Graph Execution Mode *ExecutionMode* (read only) (virtual)

Public Methods

void *GetInputs*(**List**<**Variable Definition**> inputs) (virtual)

Protected Methods

IEnumerator *Run_*(**Instruction Graph** graph, **Instruction Store** variables, int iteration)
(virtual)

<<<

DivideAssignOperator

PiRhoSoft.CompositionEngine.DivideAssignOperator : [Assign Operator](#)

Description

Constructors

DivideAssignOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (virtual)

<<<

DivideOperator

PiRhoSoft.CompositionEngine.DivideOperator : [Infix Operation](#)

Description

Static Methods

Variable Value Divide(**Variable Value** left (ref), **Variable Value** right (ref))

Constructors

DivideOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (virtual)

<<<

EnableObjectNode

PiRhoSoft.CompositionEngine.EnableObjectNode : [Instruction Graph Node](#)

Description

Constructors

EnableObjectNode()

Public Fields

Instruction Graph Node Next

Variable Reference Target

Public Properties

bool IsExecutionImmediate (read only) (virtual)

Instruction Graph Execution Mode ExecutionMode (read only) (virtual)

Public Methods

void GetInputs(**List**<**Variable Definition**> inputs) (virtual)

Protected Methods

IEnumerator Run_(**Instruction Graph** graph, **Instruction Store** variables, int iteration)
(virtual)
<<<

EqualOperator

PiRhoSoft.CompositionEngine.EqualOperator : [Infix Operation](#)

Description

Constructors

EqualOperator()
Public Methods

Variable Value Evaluate(**I Variable Store** variables) (virtual)
<<<

ExponentAssignOperator

PiRhoSoft.CompositionEngine.ExponentAssignOperator : [Assign Operator](#)

Description

Constructors

ExponentAssignOperator()
Public Methods

Variable Value Evaluate(**I Variable Store** variables) (virtual)
<<<

ExponentOperator

PiRhoSoft.CompositionEngine.ExponentOperator : [Infix Operation](#)

Description

Static Methods

Variable Value Raise(**Variable Value** left (ref), **Variable Value** right (ref))
Constructors

ExponentOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** 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

Expression Compilation Result SetStatement(string statement)

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

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

Variable Value Execute(**I Variable Store** variables)

Variable Value Execute(**I Variable Store** variables, **Variable Type** expectedType)

Variable Value Evaluate(**I Variable Store** variables)

<<<

ExpressionCommand

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

Description

Static Fields

Parameter Store *Store*

Constructors

ExpressionCommand(**Expression** expression)

Public Properties

Expression *Expression (read only)*

Public Methods

Variable Value Evaluate(**I Variable Store** 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<Expression Token> Tokenize(string input)

<<<

ExpressionNode

PiRhoSoft.CompositionEngine.ExpressionNode : [Instruction Graph Node](#)

Description

Constructors

ExpressionNode()

Public Fields

[Instruction Graph Node](#) *Next*

[Expression](#) *Expression*

Public Properties

bool *IsExecutionImmediate (read only) (virtual)*

[Instruction Graph Execution Mode](#) *ExecutionMode (read only) (virtual)*

Public Methods

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

void **GetOutputs**([List](#)<[Variable Definition](#)> outputs) *(virtual)*

Protected Methods

[IEnumerator](#) **Run_**([Instruction Graph](#) graph, [Instruction Store](#) 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, **Operator Precedence** precedence)

void AddTernaryOperator(string symbol, **Operator Precedence** precedence)

List<Operation> Parse(**List<Expression Token>** tokens)

<<<

ExpressionToken

PiRhoSoft.CompositionEngine.ExpressionToken

Description

Constructors

ExpressionToken()

Public Fields

int *Location*

Expression Token Type *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

Expression Token Type Sentinel

Expression Token Type Boolean

Expression Token Type Integer

Expression Token Type Number

Expression Token Type String

Expression Token Type Identifier

Expression Token Type Command

Expression Token Type Operator

Expression Token Type StartGroup

Expression Token Type EndGroup

Expression Token Type Separator

<<<

FloorCommand

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

Description

Constructors

FloorCommand()

Public Methods

Variable Value Evaluate(I Variable Store 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 : [Infix Operation](#)

Description

Constructors

GreaterOperator()

Public Methods

Variable Value Evaluate(I Variable Store variables) (*virtual*)

<<<

GreaterOrEqualOperator

PiRhoSoft.CompositionEngine.GreaterOrEqualOperator : [Infix Operation](#)

Description

Constructors

GreaterOrEqualOperator()

Public Methods

Variable Value Evaluate(I Variable Store variables) (*virtual*)

<<<

HideControlNode

PiRhoSoft.CompositionEngine.HideControlNode : [Instruction Graph Node](#)

Description

Constructors

HideControlNode()

Public Fields

Instruction Graph Node *Next*

Interface Reference *Control*

Public Properties

bool *IsExecutionImmediate (read only) (virtual)*

Instruction Graph Execution Mode *ExecutionMode (read only) (virtual)*

Protected Methods

IEnumerator **Run_**(**Instruction Graph** graph, **Instruction Store** variables, int iteration)
(virtual)

<<<

IIndexedVariableStore

PiRhoSoft.CompositionEngine.IIndexedVariableStore : [I Variable Store](#)

Description

Public Properties

int *Count (read only) (abstract)*

Public Methods

I Variable Store **GetItem**(int index) (abstract)

<<<

ImageBinding

PiRhoSoft.CompositionEngine.ImageBinding : [Interface Binding](#)

Description

Static Fields

string *_invalidVariableError*

string *_missingVariableError*

Constructors

ImageBinding()

Public Fields

Variable Reference *Variable*

Public Methods

void UpdateBinding(I Variable Store variables) (*virtual*)

<<<

IndexedVariableStore

PiRhoSoft.CompositionEngine.IndexedVariableStore

Description

Static Fields

string *ItemVariable*

Static Methods

Variable Value **GetVariable(I Indexed Variable Store variables, string name)**

Set Variable Result **SetVariable(I Indexed Variable Store variables, string name, Variable Value value)**

<<<

IndexedVariableStore

PiRhoSoft.CompositionEngine.IndexedVariableStore<_ItemType> : SerializedList<ItemType>, I Variable Store, I Indexed Variable Store

Description

Constructors

IndexedVariableStore()

Public Methods

I Variable Store **GetItem(int index)** (*virtual*)

Variable Value **GetVariable(string name)** (*virtual*)

Set Variable Result **SetVariable(string name, Variable Value value)** (*virtual*)

<<<

InequalOperator

PiRhoSoft.CompositionEngine.InequalOperator : Infix Operation

Description

Constructors

InequalOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** 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<[Variable Definition](#)> inputs, string source) (virtual)

Protected Constructors

InfixOperation()

<<<

Instruction

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

Description

Public Properties

bool IsRunning (read only)

bool IsExecutionImmediate (read only) (abstract)

Public Methods

[IEnumerator](#) Execute([Instruction Store](#) variables)

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

void GetOutputs(List<Variable Definition> outputs) (virtual)

Protected Constructors

Instruction()

Protected Methods

IEnumerator Run(Instruction Store variables) (abstract)

<<<

InstructionCaller

PiRhoSoft.CompositionEngine.InstructionCaller

Description

Constructors

InstructionCaller()

Public Properties

Instruction *Instruction*

List<Instruction Input> *Inputs (read only)*

List<Instruction Output> *Outputs (read only)*

bool *IsRunning (read only)*

bool *IsExecutionImmediate (read only)*

Public Methods

IEnumerator Execute(Instruction Context context, I Variable Store thisStore)

void UpdateVariables()

<<<

InstructionContext

PiRhoSoft.CompositionEngine.InstructionContext

Description

Constructors

InstructionContext()

Public Methods

void SetStore(string name, I Variable Store store)

void Clear()

<<<

InstructionDictionary

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

Description

Constructors

InstructionDictionary()

<<<

InstructionGraph

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

Description

Static Fields

string *_processFailedError*

Public Fields

Vector2 *StartPosition*

Public Properties

Instruction Graph Node List *Nodes (read only)*

bool *IsExecutionImmediate (read only) (virtual)*

Public Methods

void **GetInputs**(**List**<**Variable Definition**> inputs) (virtual)

void **GetOutputs**(**List**<**Variable Definition**> outputs) (virtual)

void **GoTo**(**Instruction Graph Node** node)

void **GoTo**(**Instruction Graph Node** node, **I Variable Store** thisStore)

void **BreakTo**(**Instruction Graph Node** node)

void **GetConnections**(**Node Data** data) (virtual)

void **SetConnection**(**Connection Data** connection, **Instruction Graph Node** target) (virtual)

Protected Constructors

InstructionGraph()

Protected Methods

IEnumerator **Run**(**Instruction Store** variables, **Instruction Graph Node** root)

<<<

InstructionGraphExecutionMode

PiRhoSoft.CompositionEngine.InstructionGraphExecutionMode

Description

Values

Instruction Graph Execution Mode *Normal*

Instruction Graph Execution Mode *Sequence*

Instruction Graph Execution Mode *Loop*

<<<

InstructionGraphNode

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

Description

Public Fields

string *Name*

Variable Reference *This*

Vector2 *GraphPosition*

Public Properties

bool *IsExecutionImmediate (read only) (abstract)*

Instruction Graph Execution Mode *ExecutionMode (read only) (abstract)*

Public Methods

void **GetInputs**(**List**<**Variable Definition**> inputs) (virtual)

void **GetOutputs**(**List**<**Variable Definition**> outputs) (virtual)

IEnumerator **Run**(**Instruction Graph** graph, **Instruction Store** variables, int executionIndex)

Color **GetNodeColor**() (virtual)

void **GetConnections**(**Node Data** data) (virtual)

void **SetConnection**(**Connection Data** connection, **Instruction Graph Node** target) (virtual)

Protected Constructors

InstructionGraphNode()

Protected Methods

IEnumerator **Run_**(**Instruction Graph** graph, **Instruction Store** variables, int iteration)
(abstract)

<<<

InstructionGraphNodeDictionary

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

Description

Constructors

InstructionGraphNodeDictionary()

<<<

InstructionGraphNodeList

PiRhoSoft.CompositionEngine.InstructionGraphNodeList : SerializedList<[Instruction Graph Node](#)>

Description

Constructors

InstructionGraphNodeList()

<<<

InstructionInput

PiRhoSoft.CompositionEngine.InstructionInput

Description

Constructors

InstructionInput()

Public Fields

[Instruction Input Type](#) *Type*

[Variable Definition](#) *Definition*

[Variable Reference](#) *Reference*

[Variable Value](#) *Value*

<<<

InstructionInputType

PiRhoSoft.CompositionEngine.InstructionInputType

Description

Values

Instruction Input Type *Reference*

Instruction Input Type *Value*

<<<

InstructionList

PiRhoSoft.CompositionEngine.InstructionList : SerializedList<[Instruction](#)>

Description

Constructors

InstructionList()

<<<

InstructionNode

PiRhoSoft.CompositionEngine.InstructionNode : [Instruction Graph Node](#)

Description

Constructors

InstructionNode()

Public Fields

Instruction Caller *Instruction*

Instruction Graph Node *Next*

bool *WaitForCompletion*

Public Properties

bool *IsExecutionImmediate (read only) (virtual)*

Instruction Graph Execution Mode *ExecutionMode (read only) (virtual)*

Protected Methods

IEnumerator **Run_**(**Instruction Graph** graph, **Instruction Store** variables, int iteration)
(virtual)

<<<

InstructionOutput

PiRhoSoft.CompositionEngine.InstructionOutput

Description

Constructors

InstructionOutput()

Public Fields

Instruction Output Type *Type*

Variable Definition *Definition*

Variable Reference *Reference*

<<<

InstructionOutputType

PiRhoSoft.CompositionEngine.InstructionOutputType

Description

Values

Instruction Output Type *Ignore*

Instruction Output Type *Reference*

<<<

InstructionStore

PiRhoSoft.CompositionEngine.InstructionStore : [I Variable Store](#)

Description

Static Fields

string *ThisStoreName*

string *SceneStoreName*

string *InputStoreName*

string *OutputStoreName*

Static Methods

bool **IsInput**([Variable Reference](#) variable)

bool **IsOutput**([Variable Reference](#) variable)

Constructors

InstructionStore(**Instruction Context** context, **I Variable Store** thisStore)

Public Properties

Instruction Context Context (read only)

I Variable Store This (read only)

I Variable Store Locals (read only)

Public Methods

void SetContext(**Instruction Context** context)

void ChangeThis(**I Variable Store** store)

void WriteInputs(**List**<**Instruction Input**> inputs)

void ReadOutputs(**List**<**Instruction Output**> outputs)

Variable Value GetVariable(string name) (virtual)

Set Variable Result SetVariable(string name, **Variable Value** value) (virtual)

<<<

IntegerVariableSource

PiRhoSoft.CompositionEngine.IntegerVariableSource : **Variable Source**<int>

Description

Constructors

IntegerVariableSource()

IntegerVariableSource(int defaultValue)

Protected Methods

bool TryGetValue(**Variable Value** variable, int value (out)) (virtual)

<<<

Interface

PiRhoSoft.CompositionEngine.Interface : **MonoBehaviour**

Description

Constructors

Interface()

Public Fields

string *Name*

Interface **Control Dictionary** *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, [I Variable Store](#) variables, string group)

Public Fields

string *BindingGroup*

Public Methods

void **UpdateBinding**([I Variable Store](#) variables) (*abstract*)

void **UpdateBinding**([I Variable Store](#) store, string group)

Protected Constructors

InterfaceBinding()

<<<

InterfaceControl

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

Description

Constructors

InterfaceControl()

Public Fields

Dependent Object List *DependentObjects*

Public Properties

I Variable Store *Variables (read only)*

bool *IsActive (read only)*

Public Methods

void *Activate()*

void *Deactivate()*

void *UpdateBindings(I Variable Store variables, string group) (virtual)*

Protected Methods

void *Setup() (virtual)*

void *Teardown() (virtual)*

<<<

InterfaceControlDictionary

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

Description

Constructors

InterfaceControlDictionary()

<<<

InterfaceManager

PiRhoSoft.CompositionEngine.InterfaceManager : SingletonBehaviour<Interface Manager>

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()**

ControlType **Activate<ControlType>()**

void **Deactivate()**

InterfaceType **GetInterface<InterfaceType>()**

ControlType **GetControl<ControlType>()**

<<<

InvertOperator

PiRhoSoft.CompositionEngine.InvertOperator : [Prefix Operation](#)

Description

Constructors

InvertOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (virtual)

<<<

ItemSelector

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

Description

Constructors

ItemSelector()

Public Properties

Selection Control 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 : [Instruction Graph Node](#)

Description

Constructors

IterateNode()

Public Fields

Instruction Graph Node *Loop*

Instruction Graph Node *Next*

Public Properties

bool *IsExecutionImmediate (read only) (virtual)*

Instruction Graph Execution Mode *ExecutionMode (read only) (virtual)*

Protected Methods

IEnumerator **Run_**(**Instruction Graph** graph, **Instruction Store** variables, int iteration)
(virtual)

<<<

IVariableList

PiRhoSoft.CompositionEngine.IVariableList

Description

Public Properties

int *VariableCount (read only) (abstract)*

Public Methods

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

Variable Value **GetVariableValue**(int index) (abstract)

Set Variable Result **SetVariableValue**(int index, **Variable Value** value) (abstract)

<<<

IVariableListener

PiRhoSoft.CompositionEngine.IVariableListener

Description

Public Methods

void **VariableChanged**(int index, **Variable Value** value) (abstract)

<<<

IVariableStore

PiRhoSoft.CompositionEngine.IVariableStore

Description

Public Methods

Variable Value `GetVariable(string name)` (*abstract*)

Set Variable Result `SetVariable(string name, Variable Value value)` (*abstract*)

<<<

LerpCommand

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

Description

Constructors

LerpCommand()

Public Methods

Variable Value `Evaluate(I Variable Store variables, string name, List<Operation> parameters)` (*virtual*)

<<<

LessOperator

PiRhoSoft.CompositionEngine.LessOperator : [Infix Operation](#)

Description

Constructors

LessOperator()

Public Methods

Variable Value `Evaluate(I Variable Store variables)` (*virtual*)

<<<

LessOrEqualOperator

PiRhoSoft.CompositionEngine.LessOrEqualOperator : [Infix Operation](#)

Description

Constructors

LessOrEqualOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (virtual)

<<<

LiteralOperation

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

Description

Constructors

LiteralOperation(Variable Value value)

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (virtual)

void ToString(StringBuilder builder) (virtual)

<<<

LocalVariableStore

PiRhoSoft.CompositionEngine.LocalVariableStore<T_> : [Variable Store](#)

Description

Public Methods

Variable Value GetVariable(string name) (virtual)

Set Variable Result SetVariable(string name, **Variable Value** value) (virtual)

Protected Constructors

LocalVariableStore(T store)

Protected Fields

T_store

<<<

LogCommand

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

Description

Constructors

LogCommand()

Public Methods

Variable Value Evaluate(**I Variable Store** variables, string name, [List<Operation>](#) parameters) (virtual)

<<<

LookupOperation

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

Description

Constructors

LookupOperation(string variable)

Public Fields

Variable Reference *Reference*

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (virtual)

void ToString(StringBuilder builder) (virtual)

void GetInputs([List<Variable Definition>](#) inputs, string source) (virtual)

<<<

LoopNode

PiRhoSoft.CompositionEngine.LoopNode : [Instruction Graph Node](#)

Description

Constructors

LoopNode()

Public Fields

Expression *Condition*

Instruction Graph Node *Loop*

Instruction Graph Node *Next*

Public Properties

bool *IsExecutionImmediate (read only) (virtual)*

Instruction Graph Execution Mode *ExecutionMode (read only) (virtual)*

Public Methods

void **GetInputs**(**List**<**Variable Definition**> inputs) *(virtual)*

Protected Methods

IEnumerator **Run_**(**Instruction Graph** graph, **Instruction Store** variables, int iteration)
(virtual)

<<<

MappedVariableStore

PiRhoSoft.CompositionEngine.MappedVariableStore : **I Variable Store**, **I Variable List**

Description

Constructors

MappedVariableStore()

Public Properties

int *VariableCount (read only) (virtual)*

Public Methods

void **Setup**(**Variable Map** map, **I Variable List**[] lists)

Variable Value **GetVariable**(string name) *(virtual)*

Set Variable Result **SetVariable**(string name, **Variable Value** value) *(virtual)*

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

Variable Value **GetVariableValue**(int index) *(virtual)*

Set Variable Result **SetVariableValue**(int index, **Variable Value** value) *(virtual)*

<<<

MaxCommand

PiRhoSoft.CompositionEngine.MaxCommand : **Command**

Description

Constructors

MaxCommand()

Public Methods

Variable Value Evaluate(**I Variable Store** 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<Variable Definition>** inputs)

string GetText(**I Variable Store** variables)

<<<

MessageBinding

PiRhoSoft.CompositionEngine.MessageBinding : [Interface Binding](#)

Description

Constructors

MessageBinding()

Public Fields

Message Message

Public Properties

TextMeshProUGUI Text (read only)

Public Methods

void UpdateBinding(I Variable Store variables) (virtual)

<<<

MessageControl

PiRhoSoft.CompositionEngine.MessageControl : [Interface Control](#)

Description

Constructors

MessageControl()

Public Fields

TextMeshProUGUI *DisplayText*

GameObject *ContinueIndicator*

GameObject *FinishedIndicator*

Public Methods

IEnumerator Show(I Variable Store variables, string text, Message Interaction Type interaction, bool isLast)

Protected Methods

IEnumerator Run(string text, Message Interaction Type interaction, bool isLast) (virtual)

void Teardown() (virtual)

void SetInteraction(Message Control Display display, Message Interaction Type interaction, bool isLast)

<<<

MessageControlDisplay

PiRhoSoft.CompositionEngine.MessageControlDisplay

Description

Values

Message Control Display *None*

Message Control Display *Continue*

Message Control Display *Finished*

<<<

MessageInteractionType

PiRhoSoft.CompositionEngine.MessageInteractionType

Description

Values

Message Interaction Type *WaitForInput*

Message Interaction Type *WaitForFinalInput*

Message Interaction Type *WaitForDisplay*

Message Interaction Type *DontWait*

<<<

MessageNode

PiRhoSoft.CompositionEngine.MessageNode : [Instruction Graph Node](#)

Description

Constructors

MessageNode()

Public Fields

Instruction Graph Node *Next*

Interface Reference *Control*

Message Interaction Type *Interaction*

bool *IsLast*

Message *Message*

Public Properties

bool *IsExecutionImmediate (read only) (virtual)*

Instruction Graph Execution Mode *ExecutionMode (read only) (virtual)*

Public Methods

void **GetInputs**(**List**<**Variable Definition**> **inputs**) *(virtual)*

Protected Methods

IEnumerator **Run_**(**Instruction Graph** **graph**, **Instruction Store** **variables**, **int** **iteration**)
(virtual)

<<<

MinCommand

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

Description

Constructors

MinCommand()

Public Methods

Variable Value Evaluate(**I Variable Store** variables, string name, [List<Operation>](#) parameters)
(virtual)

<<<

MockupConnection

PiRhoSoft.CompositionEngine.MockupConnection

Description

Constructors

MockupConnection()

Public Fields

string *Name*

Instruction Graph Node *Node*

<<<

MockupConnectionList

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

Description

Constructors

MockupConnectionList()

<<<

MockupGraph

PiRhoSoft.CompositionEngine.MockupGraph : [Instruction Graph](#)

Description

Constructors

MockupGraph()

Public Fields

Mockup Connection List *EntryPoints*

Public Methods

void GetConnections(Node Data data) *(virtual)*

void SetConnection(Connection Data connection, Instruction Graph Node target) *(virtual)*

Protected Methods

IEnumerator Run(Instruction Store variables) *(virtual)*

<<<

MockupNode

PiRhoSoft.CompositionEngine.MockupNode : [Instruction Graph Node](#)

Description

Constructors

MockupNode()

Public Fields

Mockup Connection List *Connections*

Public Properties

bool IsExecutionImmediate *(read only) (virtual)*

Instruction Graph Execution Mode *ExecutionMode (read only) (virtual)*

Public Methods

void GetConnections(Node Data data) *(virtual)*

void SetConnection(Connection Data connection, Instruction Graph Node target) *(virtual)*

Protected Methods

IEnumerator Run_(Instruction Graph graph, Instruction Store variables, int iteration)
(virtual)

<<<

ModuloAssignOperator

PiRhoSoft.CompositionEngine.ModuloAssignOperator : [Assign Operator](#)

Description

Constructors

ModuloAssignOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (virtual)

<<<

ModuloOperator

PiRhoSoft.CompositionEngine.ModuloOperator : [Infix Operation](#)

Description

Static Methods

Variable Value Modulo(**Variable Value** left (ref), **Variable Value** right (ref))

Constructors

ModuloOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (virtual)

<<<

MultiplyAssignOperator

PiRhoSoft.CompositionEngine.MultiplyAssignOperator : [Assign Operator](#)

Description

Constructors

MultiplyAssignOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (virtual)

<<<

MultiplyOperator

PiRhoSoft.CompositionEngine.MultiplyOperator : [Infix Operation](#)

Description

Static Methods

Variable Value Multiply(**Variable Value** left (*ref*), **Variable Value** right (*ref*))

Constructors

MultiplyOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (*virtual*)

<<<

NegateOperator

PiRhoSoft.CompositionEngine.NegateOperator : [Prefix Operation](#)

Description

Constructors

NegateOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (*virtual*)

<<<

NodeData

PiRhoSoft.CompositionEngine.NodeData

Description

Static Fields

float Width

float HeaderHeight

float LineHeight

float FooterHeight

Constructors

NodeData([Instruction Graph Node](#) node)

Public Fields

[List](#)<[Connection Data](#)> *Connections*

Public Properties

[Instruction Graph Node](#) *Node (read only)*

[Rect](#) *Bounds (read only)*

Public Methods

void SetPosition([Vector2](#) position)

void ClearConnections()

void AddConnections(Object obj)

void AddConnection(string name, [Instruction Graph Node](#) to)

void AddConnection(string name, string key, [Instruction Graph Node](#) to)

void AddConnection(string name, int index, [Instruction Graph Node](#) to)

<<<

NumberVariableSource

PiRhoSoft.CompositionEngine.NumberVariableSource : [Variable Source](#)<float>

Description

Constructors

NumberVariableSource()

NumberVariableSource(float defaultValue)

Protected Methods

bool TryGetValue([Variable Value](#) variable, float value (out)) (virtual)

<<<

ObjectSource

PiRhoSoft.CompositionEngine.ObjectSource

Description

Values

[Object Source](#) *Scene*

[Object Source](#) *Asset*

<<<

ObjectVariableSource

PiRhoSoft.CompositionEngine.ObjectVariableSource<_ObjectType_> : [Variable Source](#)<ObjectType>

Description

Constructors

ObjectVariableSource()

Protected Methods

bool TryGetValue([Variable Value](#) variable, [Object Type](#) value (out)) (virtual)

<<<

Operation

PiRhoSoft.CompositionEngine.Operation

Description

Static Fields

string *MismatchedMathType1Exception*

string *MismatchedMathType2Exception*

string *MismatchedBooleanType1Exception*

string *MismatchedBooleanType2Exception*

Public Methods

[Variable Value](#) Evaluate([I Variable Store](#) variables) (abstract)

void ToString(StringBuilder builder) (abstract)

void GetInputs([List](#)<[Variable Definition](#)> inputs, string source) (virtual)

void GetOutputs([List](#)<[Variable Definition](#)> outputs, string source) (virtual)

string ToString() (virtual)

Protected Constructors

Operation()

<<<

OperatorPrecedence

PiRhoSoft.CompositionEngine.OperatorPrecedence : ValueType

Description

Static Fields

Operator Precedence *Assignment*

Operator Precedence *Or*

Operator Precedence *And*

Operator Precedence *Ternary*

Operator Precedence *Equality*

Operator Precedence *Comparison*

Operator Precedence *Addition*

Operator Precedence *Multiplication*

Operator Precedence *Exponentiation*

Static Methods

Operator Precedence **LeftAssociative(int value)**

Operator Precedence **RightAssociative(int value)**

Public Properties

int *Value (read only)*

int *AssociativeValue (read only)*

<<<

OrAssignOperator

PiRhoSoft.CompositionEngine.OrAssignOperator : [Assign Operator](#)

Description

Constructors

OrAssignOperator()

Public Methods

Variable Value **Evaluate(I Variable Store variables)** *(virtual)*

<<<

OrOperator

PiRhoSoft.CompositionEngine.OrOperator : [Infix Operation](#)

Description

Static Methods

Variable Value Or(**Variable Value** left (*ref*), **Variable Value** right (*ref*))

Constructors

OrOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (*virtual*)

<<<

ParameterStore

PiRhoSoft.CompositionEngine.ParameterStore : **I Variable Store**

Description

Static Fields

string *ParameterName*

Constructors

ParameterStore()

Public Fields

List<Variable Value> *Parameters*

Public Methods

Variable Value GetVariable(string name) (*virtual*)

Set Variable Result SetVariable(string name, **Variable Value** value) (*virtual*)

<<<

PlayAnimation

PiRhoSoft.CompositionEngine.PlayAnimation : **Instruction Graph Node**

Description

Constructors

PlayAnimation()

Public Fields

[Instruction Graph Node](#) *Next*

[Variable Source Type](#) *Type*

[Variable Reference](#) *Target*

[AnimationClip](#) *Animation*

[Variable Reference](#) *AnimationReference*

bool *WaitForCompletion*

Public Properties

bool *IsExecutionImmediate (read only) (virtual)*

[Instruction Graph Execution Mode](#) *ExecutionMode (read only) (virtual)*

Public Methods

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

Protected Methods

[IEnumerator](#) **Run_**([Instruction Graph](#) graph, [Instruction Store](#) variables, int iteration)
(virtual)

<<<

PlayAnimationState

PiRhoSoft.CompositionEngine.PlayAnimationState : [Instruction Graph Node](#)

Description

Constructors

PlayAnimationState()

Public Fields

[Instruction Graph Node](#) *Next*

[Variable Source Type](#) *Type*

[Variable Reference](#) *Target*

string *State*

[Variable Reference](#) *StateReference*

Public Properties

bool *IsExecutionImmediate (read only) (virtual)*

[Instruction Graph Execution Mode](#) *ExecutionMode (read only) (virtual)*

Public Methods

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

Protected Methods

IEnumerator Run_(**Instruction Graph** graph, **Instruction Store** variables, int iteration)

(virtual)

<<<

PowCommand

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

Description

Constructors

PowCommand()

Public Methods

Variable Value Evaluate(**I Variable Store** 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<Variable Definition>](#) inputs, string source) (virtual)

Protected Constructors

PrefixOperation()

<<<

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_> : [I Variable List](#)

Description

Constructors

PropertyList(**OwnerType** owner, [Property Map](#)<**OwnerType**> map)

Public Properties

int *VariableCount* (read only) (virtual)

Public Methods

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

[Variable Value](#) **GetVariableValue**(**int** index) (virtual)

[Set Variable Result](#) **SetVariableValue**(**int** index, [Variable Value](#) value) (virtual)

<<<

PropertyMap

PiRhoSoft.CompositionEngine.PropertyMap

Description

Public Properties

int *PropertyCount* (read only) (abstract)

Public Methods

string **GetPropertyName**(**int** index) (abstract)

[I Variable List](#) **CreateList**(**Object** owner) (abstract)

Protected Constructors

PropertyMap()

<<<

PropertyMap

PiRhoSoft.CompositionEngine.PropertyMap<_OwnerType_> : [Property Map](#)

Description

Constructors

PropertyMap()

Public Fields

[List](#)<[Property](#)<[OwnerType](#)>> *Properties*

Public Properties

int *PropertyCount (read only) (virtual)*

Public Methods

string *GetPropertyName(int index) (virtual)*

I Variable List *CreateList(Object owner) (virtual)*

[Property Map](#)<_OwnerType_> **Add**(string name, Func<[OwnerType](#), [OwnerType](#)> getter, Func<[OwnerType](#), [OwnerType](#), [OwnerType](#)> setter)

<<<

RandomCommand

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

Description

Constructors

RandomCommand()

Public Methods

Variable Value **Evaluate**(**I Variable Store** variables, string name, [List](#)<[Operation](#)> parameters) *(virtual)*

<<<

RoundCommand

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

Description

Constructors

RoundCommand()

Public Methods

Variable Value Evaluate(**I Variable Store** variables, string name, **List<Operation>** parameters) (virtual)

<<<

SceneVariableStore

PiRhoSoft.CompositionEngine.SceneVariableStore : **I Variable Store**

Description

Constructors

SceneVariableStore()

Public Methods

Variable Value GetVariable(string name) (virtual)

Set Variable Result SetVariable(string name, **Variable Value** value) (virtual)

<<<

ScopedGraph

PiRhoSoft.CompositionEngine.ScopedGraph : **Instruction Graph**

Description

Constructors

ScopedGraph()

Public Fields

Instruction Graph Node Enter

Instruction Graph Node Process

Instruction Graph Node Exit

Protected Methods

IEnumerator Run(**Instruction Store** variables) (virtual)

<<<

ScrollSelection

PiRhoSoft.CompositionEngine.ScrollSelection : [Selection Control](#)

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

Selection Control *SelectionControl*

string *BindingGroup*

<<<

SelectionControl

PiRhoSoft.CompositionEngine.SelectionControl : [Interface Control](#)

Description

Constructors

SelectionControl()

Public Fields

bool *VerticalWrapping*

bool *HorizontalWrapping*

Public Properties

Selection Item *FocusedItem (read only)*

I Variable Store *FocusedVariables (read only)*

Selection Item *SelectedItem (read only)*

I Variable Store *SelectedVariables (read only)*

Public Methods

void **UpdateBindings**(**I Variable Store** variables, string group) (*virtual*)

IEnumerator **MakeSelection**(**I Variable Store** variables, **IEnumerable**<**Selection Item**> 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*

bool *_columnMajor*

int *_columnIndex*

int *_rowIndex*

List<**Menu Item**> *_items*

Menu Item *_focusedItem*

Menu Item *_selectedItem*

Protected Methods

void Teardown() *(virtual)*

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

Menu Item **GetItem(int column, int row)**

void FocusItem(Menu Item item) *(virtual)*

void BlurItem(Menu Item 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 : [I Variable Store](#)

Description

Constructors

SelectionItem()

Public Fields

string *Label*

Variable Reference *Item*

Object Source *Source*

GameObject *Template*

bool *Expand*

Public Methods

Variable Value **GetVariable(string name)** *(virtual)*

Set Variable Result **SetVariable(string name, Variable Value value)** *(virtual)*

<<<

SelectionNode

PiRhoSoft.CompositionEngine.SelectionNode : [Instruction Graph Node](#)

Description

Constructors

SelectionNode()

Public Fields

[Instruction Graph Node](#) *OnCanceled*

[Interface Reference](#) *Control*

bool *IsSelectionRequired*

[Selection Node Item List](#) *Items*

Public Properties

bool *IsExecutionImmediate (read only) (virtual)*

[Instruction Graph Execution Mode](#) *ExecutionMode (read only) (virtual)*

Public Methods

void **GetConnections**([Node Data](#) data) *(virtual)*

void **SetConnection**([Connection Data](#) connection, [Instruction Graph Node](#) target) *(virtual)*

Protected Methods

[IEnumerator](#) **Run_**([Instruction Graph](#) graph, [Instruction Store](#) variables, int iteration)
(virtual)

<<<

SelectionNodeItem

PiRhoSoft.CompositionEngine.SelectionNodeItem : [Selection Item](#)

Description

Constructors

SelectionNodeItem()

Public Fields

[Instruction Graph Node](#) *OnSelected*

<<<

SelectionNodeItemList

PiRhoSoft.CompositionEngine.SelectionNodeItemList : SerializedList<[Selection Node Item](#)>

Description

Constructors

SelectionNodeItemList()

<<<

SequenceNode

PiRhoSoft.CompositionEngine.SequenceNode : [Instruction Graph Node](#)

Description

Constructors

SequenceNode()

Public Fields

[Instruction Graph Node List](#) *Sequence*

[Instruction Graph Node](#) *Next*

Public Properties

bool *IsExecutionImmediate (read only) (virtual)*

[Instruction Graph Execution Mode](#) *ExecutionMode (read only) (virtual)*

Protected Methods

[IEnumerator](#) **Run_**([Instruction Graph](#) graph, [Instruction Store](#) variables, int iteration)
(virtual)

<<<

SerializedVariable

PiRhoSoft.CompositionEngine.SerializedVariable

Description

Constructors

SerializedVariable()

Public Fields

string *Name*

Variable Type *Type*

string *Data*

Object *Object*

Public Methods

void SetVariable(**Variable** variable)

void SetValue(**Variable Value** value)

Variable GetVariable()

Variable Value GetValue()

<<<

SetAnimationParameter

PiRhoSoft.CompositionEngine.SetAnimationParameter : [Instruction Graph Node](#)

Description

Constructors

SetAnimationParameter()

Public Fields

Instruction Graph Node *Next*

Variable Reference *Target*

string *Parameter*

AnimatorControllerParameterType *Type*

bool *BoolValue*

int *IntValue*

float *FloatValue*

Public Properties

bool *IsExecutionImmediate (read only) (virtual)*

Instruction Graph Execution Mode *ExecutionMode (read only) (virtual)*

Public Methods

void GetInputs(**List**<**Variable Definition**> inputs) *(virtual)*

Protected Methods

IEnumerator Run_(**Instruction Graph** graph, **Instruction Store** variables, int iteration)
(virtual)

<<<

SetVariableResult

PiRhoSoft.CompositionEngine.SetVariableResult

Description

Values

Set Variable Result *Success*

Set Variable Result *NotFound*

Set Variable Result *ReadOnly*

Set Variable Result *TypeMismatch*

<<<

ShowControlNode

PiRhoSoft.CompositionEngine.ShowControlNode : [Instruction Graph Node](#)

Description

Constructors

ShowControlNode()

Public Fields

Instruction Graph Node *Next*

Interface Reference *Control*

Public Properties

bool *IsExecutionImmediate (read only) (virtual)*

Instruction Graph Execution Mode *ExecutionMode (read only) (virtual)*

Protected Methods

IEnumerator **Run_**(**Instruction Graph** graph, **Instruction Store** variables, int iteration)
(virtual)

<<<

SignCommand

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

Description

Constructors

SignCommand()

Public Methods

Variable Value Evaluate(**I Variable Store** variables, string name, **List<Operation>** parameters) (virtual)

<<<

SimpleGraph

PiRhoSoft.CompositionEngine.SimpleGraph : [Instruction Graph](#)

Description

Constructors

SimpleGraph()

Public Fields

Instruction Graph Node *Process*

Protected Methods

IEnumerator Run(**Instruction Store** variables) (virtual)

<<<

SinCommand

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

Description

Constructors

SinCommand()

Public Methods

Variable Value Evaluate(**I Variable Store** variables, string name, **List<Operation>** parameters) (virtual)

<<<

SqrtCommand

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

Description

Constructors

SqrtCommand()

Public Methods

Variable Value Evaluate(**I Variable Store** variables, string name, **List<Operation>** parameters) (virtual)

<<<

StringVariableSource

PiRhoSoft.CompositionEngine.StringVariableSource : **Variable Source**<string>

Description

Constructors

StringVariableSource()

StringVariableSource(string defaultValue)

Protected Methods

bool TryGetValue(**Variable Value** variable, string value (out)) (virtual)

<<<

SubtractAssignOperator

PiRhoSoft.CompositionEngine.SubtractAssignOperator : **Assign Operator**

Description

Constructors

SubtractAssignOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (virtual)

<<<

SubtractOperator

PiRhoSoft.CompositionEngine.SubtractOperator : **Infix Operation**

Description

Static Methods

Variable Value Subtract(**Variable Value** left (*ref*), **Variable Value** right (*ref*))

Constructors

SubtractOperator()

Public Methods

Variable Value Evaluate(**I Variable Store** variables) (*virtual*)

<<<

TanCommand

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

Description

Constructors

TanCommand()

Public Methods

Variable Value Evaluate(**I Variable Store** 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)*

Variable Value Evaluate(**I Variable Store** variables) *(virtual)*

<<<

TextBinding

PiRhoSoft.CompositionEngine.TextBinding : [Interface Binding](#)

Description

Constructors

TextBinding()

Public Fields

Variable Reference *Variable*

Public Methods

void UpdateBinding(I Variable Store variables) *(virtual)*

<<<

TruncateCommand

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

Description

Constructors

TruncateCommand()

Public Methods

Variable Value Evaluate(**I Variable Store** variables, string name, [List<Operation>](#) parameters) *(virtual)*

<<<

UpdateBindingNode

PiRhoSoft.CompositionEngine.UpdateBindingNode : [Instruction Graph Node](#)

Description

Constructors

UpdateBindingNode()

Public Fields

[Instruction Graph Node](#) *Next*

[Interface Reference](#) *Control*

string *Bindings*

Public Properties

bool *IsExecutionImmediate (read only) (virtual)*

[Instruction Graph Execution Mode](#) *ExecutionMode (read only) (virtual)*

Protected Methods

IEnumerator **Run_**([Instruction Graph](#) graph, [Instruction Store](#) variables, int iteration)
(virtual)

<<<

Variable

PiRhoSoft.CompositionEngine.Variable : ValueType

Description

Static Properties

[Variable](#) *Empty (read only)*

Static Methods

[Variable](#) **Create**(string name, [Variable Value](#) value)

Public Properties

string *Name (read only)*

[Variable Value](#) *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

Variable Definition Create(string name, **Variable Type** type, string availability, **Expression** initializer)

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

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

Variable Definition Create(string name, string values, string availability, **Expression** initializer)

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

Variable Definition Create(string name, **Variable Type** type, bool constrainRange, float minimum, float maximum, string typeConstraint, string availability, **Expression** initializer)

Public Properties

string *Name (read only)*

Variable Type *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(**I Variable Store** variables)

<<<

VariableInitializerAttribute

PiRhoSoft.CompositionEngine.VariableInitializerAttribute : Attribute

Description

Constructors

VariableInitializerAttribute(**Variable Initializer Type** type)

Public Fields

Variable Initializer Type *Type*

<<<

VariableInitializerType

PiRhoSoft.CompositionEngine.VariableInitializerType

Description

Values

Variable Initializer Type *Expression*

Variable Initializer Type *DefaultValue*

Variable Initializer Type *None*

<<<

VariableList

PiRhoSoft.CompositionEngine.VariableList : **I Variable List**

Description

Constructors

VariableList()

Public Properties

Variable Schema *Schema (read only)*

I Variable Store *Owner (read only)*

bool *NeedsUpdate (read only)*

int *VariableCount (read only) (virtual)*

Public Methods

```

void LoadFrom(Variable List variables, string availability)
void SaveTo(Variable List variables, string availability)
void Setup(Variable Schema schema, I Variable Store owner)
void Update()
void Reset(int index)
void Reset(string availability)
void Reset(IList<string> variables)
void Clear()
string GetVariableName(int index) (virtual)
Variable Value GetVariableValue(int index) (virtual)
Set Variable Result SetVariableValue(int index, Variable Value value) (virtual)
<<<

```

VariableListener

PiRhoSoft.CompositionEngine.VariableListener : **I Variable List**

Description

Constructors

```

VariableListener(I Variable Listener listener, I Variable List list)
    # Public Properties

int VariableCount (read only) (virtual)
    # Public Methods

string GetVariableName(int index) (virtual)
Variable Value GetVariableValue(int index) (virtual)
Set Variable Result SetVariableValue(int index, Variable Value 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))
Variable Map Add(Variable Schema schema)
Variable Map Add(Property Map 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)
Variable Value GetValue(I Variable Store variables)
Set Variable Result SetValue(I Variable Store variables, Variable Value value)
string ToString() (virtual)
<<<

```

VariableSchema

PiRhoSoft.CompositionEngine.VariableSchema

Description

Constructors

```

VariableSchema()
    # Public Properties

```

int *Version (read only)*

int *Count (read only)*

Variable Definition *Item*

Public Methods

int **GetIndex**(string name)

bool **HasDefinition**(string name)

bool **AddDefinition**(string name, **Variable Type** type)

void **RemoveDefinition**(int index)

<<<

VariableSource

PiRhoSoft.CompositionEngine.VariableSource<_T_>

Description

Public Fields

Variable Source Type *Type*

Variable Reference *Reference*

T *Value*

Public Methods

void **GetInputs**(**List**<**Variable Definition**> inputs)

bool **TryGetValue**(**I Variable Store** variables, **Object** context, **T** value (out))

Protected Constructors

VariableSource()

Protected Methods

bool **TryGetValue**(**Variable Value** variable, **T** Value (out)) (abstract)

<<<

VariableSourceType

PiRhoSoft.CompositionEngine.VariableSourceType

Description

Values

Variable Source Type *Value*

Variable Source Type *Reference*

<<<

VariableStore

PiRhoSoft.CompositionEngine.VariableStore : [I Variable Store](#)

Description

Constructors

VariableStore()

Public Properties

[List<Variable>](#) *Variables (read only)*

[Dictionary<string, string>](#) *Map (read only)*

Public Methods

[Variable Value](#) **GetVariable(string name)** *(virtual)*

[Set Variable Result](#) **SetVariable(string name, [Variable Value](#) value)** *(virtual)*

void Clear()

<<<

VariableType

PiRhoSoft.CompositionEngine.VariableType

Description

Values

[Variable Type](#) *Empty*

[Variable Type](#) *Boolean*

[Variable Type](#) *Integer*

[Variable Type](#) *Number*

[Variable Type](#) *String*

[Variable Type](#) *Object*

[Variable Type](#) *Store*

<<<

VariableValue

PiRhoSoft.CompositionEngine.VariableValue : ValueType, IComparable<string>, IComparable<float>, IComparable<int>, IComparable<bool>, IComparable<[Variable Value](#)>, IEquatable<string>, IEquatable<float>, IEquatable<int>, IEquatable<bool>, IEquatable<[Variable](#)

Value>

Description

Static Fields

Variable Value *Empty*

Static Methods

Variable Type GetType(Type type)

Variable Value Create(Variable Type type)

Variable Value Create<T>(T value)

Public Properties

Variable Type Type (read only)

bool Boolean (read only)

int Integer (read only)

float Number (read only)

string String (read only)

Object Object (read only)

I Variable Store 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(I Variable Store variables (out))

bool Equals(Variable Value 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(I Variable Store value)

bool Equals(Object other) (virtual)

int CompareTo(Variable Value 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(I Variable Store store)
int GetHashCode() (virtual)
string Write()
void Read(string value)
    <<<

```

VisibleBinding

PiRhoSoft.CompositionEngine.VisibleBinding : [Interface Binding](#)

Description

Constructors

VisibleBinding()

Public Fields

[Expression](#) *Condition*

Public Methods

void UpdateBinding(**I Variable Store** variables) *(virtual)*

<<<

WaitNode

PiRhoSoft.CompositionEngine.WaitNode : [Instruction Graph Node](#)

Description

Constructors

WaitNode()

Public Fields

[Instruction Graph Node](#) *Next*

float *Time*

Public Properties

bool *IsExecutionImmediate (read only) (virtual)*

[Instruction Graph Execution Mode](#) *ExecutionMode (read only) (virtual)*

Protected Methods

IEnumerator Run_(**Instruction Graph** graph, **Instruction Store** variables, int iteration)
(virtual)