

Table of Contents

Inventory Asset Documentation

Inventory System

Data

Inventory

InventoryItem

InventoryType

KeyItem

SortByEnum

SpawnItem

TransferItem

Editor

InventoryWindow

Script

ContextMenuGenerator

CreateElements

CustomText

DemoSceneManager

DictionaryActions

Enemy

EnemyList

EquippedItem

GenerateData

InventoryManager

InventorySpawner

InventoryTarget

InventoryTrader

PickupItem

PickupUI

Player

Storage

Trader

UI

IconElement

IconElement.UxmlFactory

InventoryElement

InventoryElement.UxmlFactory

SlotElement

SlotElement.UxmlFactory

SlotPopupElement

SlotPopupElement.UxmlFactory

TransferPopupElement

TransferPopupElement.UxmlFactory

Item System

Data

ItemData

ItemType

WeaponState

Editor

CreateItem

EditItem

ItemIDScrubber

Script

WorldItem

Utility System

Data

ContextMenuIconButtonData

ContextMenuData

ContextMenuItemParentData

UIAnchorEnum

Extension

FloatExtension

IntExtension

TransformExtension

VectorExtension

VisualElementExtension

VisualElementStyleExtension

Script

Assets

CreateElements

Movement

RichText

Tool

Timer

UI

ContextMenuElement

ContextMenuElement.UxmlFactory

ContextMenuItemButtonElement

ContextMenuItemButtonElement.UxmlFactory

ContextMenuItemElement

ContextMenuItemElement.UxmlFactory

ContextMenuItemParentElement

ContextMenuItemParentElement.UxmlFactory

DialogBoxElement

DialogBoxElement.UxmlFactory

PopupBaseElement

SpriteElement

SpriteElement.UxmlFactory

ToolTipElement

ToolTipElement.UxmlFactory

Namespace InventorySystem

This is The Namespace for the InventorySystem, Assembly-CSharp

InventorySystem.Data

InventorySystem.Editor

InventorySystem.Script

InventorySystem.UI

Namespace InventorySystem.Data

Classes

[Inventory](#)

This class is a Scriptable Object that stores the values from an inventory.

[InventoryItem](#)

This class is the Item for the Inventory.

[KeyItem](#)

This class is the KeyItem it is used for storage the InventoryItem and its key for the Dictionary. and is used to convert for saving string

[SpawnItem](#)

This class is the SpawnItem that is used to create an InventoryItem for the inventory system.

[TransferItem](#)

Enums

[InventoryType](#)

This is used to determine how an Inventory should be Interacted with. To use this enum in UI Builder

[SortByEnum](#)

This is used to determine how an Inventory will be sorted. To use this enum in UI Builder

Class Inventory

This class is a Scriptable Object that stores the values from an inventory.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.ScriptableObject
Inventory

Namespace: [InventorySystem.Data](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[CreateAssetMenu]
[Serializable]
public class Inventory : ScriptableObject
```

Fields

currencyChanged

This action will be invoked when the inventory's currency is changed.

Declaration

```
public Action<int, int> currencyChanged
```

Field Value

TYPE	DESCRIPTION
System.Action<System.Int32, System.Int32>	

Remarks

Pass the currencyValue variable & the value it was changed by.

maxWeight

This is the maximum weight of the inventory.

Declaration

```
public float maxWeight
```

Field Value

TYPE	DESCRIPTION
System.Single	

Storage

This is where the contents of the inventory is held.

Declaration

```
[SerializeField]  
public List<KeyItem> Storage
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<KeyItem>	

Properties

CurrencyValue

This is the amount of gold that is held.

Declaration

```
[SerializeField]  
public int CurrencyValue { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	When the value changes it will Invoke the currencyChanged Action. & pass the currencyValue variable & the value it was changed by.

FreeSlots

This will show how many slots are left in this inventory.

Declaration

```
public int FreeSlots { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

IsTrader

This will check if this Inventory is of type 'InventoryType.Trader'.

Declaration

```
public bool IsTrader { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Size

This is the Size of the inventory.

Declaration

```
[SerializeField]  
public int Size { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	This will Update the Storage when changed.

TitleName

This is the name that is used in te UI title.

Declaration

```
public string TitleName { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Type

This is the type of inventory this is.

Declaration

```
public InventoryType Type { get; set; }
```

Property Value

TYPE	DESCRIPTION
InventoryType	

UID

This is the unique id for the inventory.

Declaration

```
public string UID { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

ClearCurrency()

This will clear the currency of the attached inventory.

Declaration

```
public void ClearCurrency()
```

DecreaseCurrency(Int32)

This will decrease the currency of the attached inventory.

Declaration

```
public void DecreaseCurrency(int money)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	money	This is how much will be removed to the inventory's currency.

GetCurrency()

This will get the currency of the attached inventory.

Declaration

```
public int GetCurrency()
```

Returns

TYPE	DESCRIPTION
System.Int32	The return is the amount of currency the inventory has.

IncreaseCurrency(Int32)

This will increase the currency of the attached inventory.

Declaration

```
public void IncreaseCurrency(int money)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	money	This is how much will be added to the inventory's currency.

Init()

This is a an Init and will set a new UID;

Declaration

```
public void Init()
```

Init(Inventory)

This is an Init that will populates from another Inventory File.

Declaration

```
public void Init(Inventory inventory)
```

Parameters

TYPE	NAME	DESCRIPTION
Inventory	inventory	This param Inventory will add it's contents to this Inventory

Init(String)

This is a an Init and will set a new UID; and the TitleName

Declaration

```
public void Init(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	This is the TitleName of the inventory.

Init(String, Int32, InventoryType)

This is a an Init and will set a new UID; and the TitleName. and the inventory size. and the type of inventory it is.

Declaration

```
public void Init(string name, int size, InventoryType type)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	This is the TitleName of the inventory.
System.Int32	size	This is the size of the inventory capacity.
InventoryType	type	This is the type of inventory it is.

InventoryWeight()

This will get the current weight of the inventory items.

Declaration

```
public float InventoryWeight()
```

Returns

TYPE	DESCRIPTION
System.Single	The return is the current weight of all of the items in the inventory.

Extension Methods

- [RichText.UpperCaseText\(Object\)](#)
- [RichText.LowerCaseText\(Object\)](#)
- [RichText.BoldText\(Object\)](#)
- [RichText.ItalicText\(Object\)](#)
- [RichText.Text\(Object\)](#)
- [RichText.UnderLineText\(Object\)](#)
- [RichText.StrikeText\(Object\)](#)
- [RichText.SupText\(Object\)](#)
- [RichText.SubText\(Object\)](#)
- [RichText.PositionText\(Object, Single\)](#)
- [RichText.SizeText\(Object, Int32\)](#)
- [RichText.ColoredText\(Object, String\)](#)
- [RichText.ColoredText\(Object, Color\)](#)
- [RichText.BooleanText\(Object, Boolean\)](#)
- [RichText.BooleanText\(Object, Boolean, Color\)](#)

Class InventoryItem

This class is the Item for the Inventory.

Inheritance

System.Object
InventoryItem

Namespace: [InventorySystem.Data](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class InventoryItem
```

Constructors

InventoryItem(InventoryItem)

This constructor Populates the InventoryItem File with data from another InventoryItem File

Declaration

```
public InventoryItem(InventoryItem itemInventory)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryItem	itemInventory	Copy the value from the param to this InventoryItem.

InventoryItem(ItemData, Int32)

This populates the InventoryItem File with data from an ItemData File.

Declaration

```
public InventoryItem(ItemData itemData, int currentStack = 1)
```

Parameters

TYPE	NAME	DESCRIPTION
ItemData	itemData	Copy the value from the param to this InventoryItem.

TYPE	NAME	DESCRIPTION
System.Int32	currentStack	Copy the param to this InventoryItem.currentStack.

Fields

currentStack

This is the amount of the item.

Declaration

```
public int currentStack
```

Field Value

TYPE	DESCRIPTION
System.Int32	

isEquipped

This is used to determine if this item is equipped.

Declaration

```
public bool isEquipped
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

itemData

The data of the item.

Declaration

```
public ItemData itemData
```

Field Value

TYPE	DESCRIPTION
ItemData	

Properties

ItemFreeSpace

This is how much freeSpace this item has. (how many more items that can be added to the currentStack)

Declaration

```
public int ItemFreeSpace { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

ItemWeight

This is how much the item weights times the number of the current stack size.

Declaration

```
public float ItemWeight { get; }
```

Property Value

TYPE	DESCRIPTION
System.Single	

TotalCost

This is the totalCost of all of the item(s) in the currentStack.

Declaration

```
public int TotalCost { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Methods

AmountCanBuy(Int32)

This will see how may of this item can be purchased.

Declaration

```
public int AmountCanBuy(int currencyValue)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	currencyValue	This is the available currency that will see how many can be purchased.

Returns

TYPE	DESCRIPTION
System.Int32	The return is the amount that can be purchased. (Round Down)

CombineIntoItem(InventoryItem)

This will combine this item's current stack into the param item's currentstack.

Declaration

```
public bool CombineIntoItem(InventoryItem itemInventory)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryItem	itemInventory	This param item will be fill with the item that called this method.

Returns

TYPE	DESCRIPTION
System.Boolean	return true if any amount has been transferred.

DropItem(Transform)

This will create a World Object from the ItemData foreach item in currentStack.

Declaration

```
public void DropItem(Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	

DropItem(Transform, Int32)

This will create a World Object from the ItemData for amount.

Declaration

```
public void DropItem(Transform transform, int amount)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	
System.Int32	amount	how many of the item to drop clamped between 1 and currentStack.

DropItemIndividual(Transform)

This will create a World Objects for every item in currentStack.

Declaration

```
public void DropItemIndividual(Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	

DropItemIndividual(Transform, Int32)

This will create a World Objects for every item in amount.

Declaration

```
public void DropItemIndividual(Transform transform, int amount)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	
System.Int32	amount	how many of the item to drop clamped between 1 and currentStack.

EquipItem()

This will create a World Object from the ItemData foreach item in currentStack.

Declaration

```
public void EquipItem()
```

SplitItem(Int32)

This will create a new item and give the amount from this item's currentStack to the new item.

Declaration

```
public InventoryItem SplitItem(int amount)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	amount	This is the amount that will be split into the new item.

Returns

TYPE	DESCRIPTION
InventoryItem	The return is the new item.

TradeStackAmount(InventoryItem, Int32)

This will trade the stack from this item to the param item, by param amount.

Declaration

```
public void TradeStackAmount(InventoryItem target, int amount)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryItem	target	This item will receive the amount.
System.Int32	amount	This is the amount to be traded.

UnEquipItem()

This will unequipped item and destroy the World Object.

Declaration

```
public void UnequipItem()
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)

Enum InventoryType

This is used to determine how an Inventory should be Interacted with. To use this enum in UI Builder

Namespace: [InventorySystem.Data](#)

Assembly: [Assembly-CSharp.dll](#)

Syntax

```
public enum InventoryType
```

Remarks

InventorySystem.Data.InventoryType, Assembly-CSharp

Fields

NAME	DESCRIPTION
Loot	
Player	
Storage	
Trader	

Extension Methods

- [RichText.UpperCaseText\(\)](#)
- [RichText.LowerCaseText\(\)](#)
- [RichText.BoldText\(\)](#)
- [RichText.ItalicText\(\)](#)
- [RichText.Text\(\)](#)
- [RichText.UnderLineText\(\)](#)
- [RichText.StrikeText\(\)](#)
- [RichText.SupText\(\)](#)
- [RichText.SubText\(\)](#)
- [RichText.PositionText\(Single\)](#)
- [RichText.SizeText\(Int32\)](#)
- [RichText.ColoredText\(String\)](#)
- [RichText.ColoredText\(Color\)](#)
- [RichText.BooleanText\(Boolean\)](#)
- [RichText.BooleanText\(Boolean, Color\)](#)

Class KeyItem

This class is the KeyItem it is used for storage the InventoryItem and its key for the Dictionary. and is used to convert for saving string

Inheritance

System.Object
KeyItem

Namespace: [InventorySystem.Data](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class KeyItem
```

Constructors

KeyItem(Int32, InventoryItem)

This will take a Key and InventoryItem.

Declaration

```
public KeyItem(int key, InventoryItem item)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	key	This integer will be the key for the dictionary Key.
InventoryItem	item	this InventoryItem will be the value for the dictionary Value.

KeyItem(String)

This will accept a string with the 'ConvertToStorage' format "Key:Item.IID:ItemCurrentStack". and convert the values into the Key and Item

Declaration

```
public KeyItem(string storage)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	storage	This param is the string that should be formatted correctly.

Properties

ConvertToStorage

This will convert the KeyItem into a strong for saving.

Declaration

```
public string ConvertToStorage { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	The return will be a string for saving and storage.

Item

This is the Item that is the pair value of the key in the dictionary.

Declaration

```
public InventoryItem Item { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryItem	

Key

This is the Key of the item in its owner's dictionary

Declaration

```
public int Key { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Extension Methods

- RichText.UpperCaseText(Object)
- RichText.LowerCaseText(Object)
- RichText.BoldText(Object)
- RichText.ItalicText(Object)
- RichText.Text(Object)
- RichText.UnderLineText(Object)
- RichText.StrikeText(Object)
- RichText.SupText(Object)
- RichText.SubText(Object)
- RichText.PositionText(Object, Single)
- RichText.SizeText(Object, Int32)
- RichText.ColoredText(Object, String)
- RichText.ColoredText(Object, Color)
- RichText.BooleanText(Object, Boolean)
- RichText.BooleanText(Object, Boolean, Color)

Enum SortByEnum

This is used to determine how an Inventory will be sorted. To use this enum in UI Builder

Namespace: [InventorySystem.Data](#)

Assembly: [Assembly-CSharp.dll](#)

Syntax

```
public enum SortByEnum
```

Remarks

InventorySystem.Data.SortByEnum, Assembly-CSharp

Fields

NAME	DESCRIPTION
None	
SortByAmount	
SortByCost	
SortByName	
SortByType	
SortByWeight	

Extension Methods

- [RichText.UpperCaseText\(\)](#)
- [RichText.LowerCaseText\(\)](#)
- [RichText.BoldText\(\)](#)
- [RichText.ItalicText\(\)](#)
- [RichText.Text\(\)](#)
- [RichText.UnderLineText\(\)](#)
- [RichText.StrikeText\(\)](#)
- [RichText.SupText\(\)](#)
- [RichText.SubText\(\)](#)
- [RichText.PositionText\(Single\)](#)
- [RichText.SizeText\(Int32\)](#)
- [RichText.ColoredText\(String\)](#)
- [RichText.ColoredText\(Color\)](#)
- [RichText.BooleanText\(Boolean\)](#)
- [RichText.BooleanText\(Boolean, Color\)](#)

Class SpawnItem

This class is the SpawnItem that is used to create an InventoryItem for the inventory system.

Inheritance

System.Object
SpawnItem

Namespace: [InventorySystem.Data](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class SpawnItem
```

Constructors

SpawnItem(ItemData)

This constructor will create a SpawnItem that will be used to create an InventoryItem.

Declaration

```
public SpawnItem(ItemData item)
```

Parameters

TYPE	NAME	DESCRIPTION
ItemData	item	This is the ItemData that will give the InventoryItem its data.

SpawnItem(ItemData, Int32, Boolean)

This constructor will create a SpawnItem that will be used to create an InventoryItem.

Declaration

```
public SpawnItem(ItemData item, int amount, bool isRandom = false)
```

Parameters

TYPE	NAME	DESCRIPTION
ItemData	item	This is the ItemData that will give the InventoryItem its data.
System.Int32	amount	This is the Amount that will be added to the InventoryItem.

TYPE	NAME	DESCRIPTION
System.Boolean	isRandom	This will determine if the Amount will be used or a random number.

Properties

Amount

This is the Amount that will be added to the InventoryItem.

Declaration

```
public int Amount { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

CreateItem

This will Create an InventoryItem from it's data.

Declaration

```
public InventoryItem CreateItem { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryItem	The return will be the InventoryItem that will be created.

IsRandom

This will determine if the amount will be used for a random number from (1 to the Item.maxStack) is used.

Declaration

```
public bool IsRandom { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Item

This is the Data that is going to be used to create the InventoryItem.

Declaration

```
public ItemData Item { get; }
```

Property Value

TYPE	DESCRIPTION
ItemData	

Extension Methods

- RichText.UpperCaseText(Object)
- RichText.LowerCaseText(Object)
- RichText.BoldText(Object)
- RichText.ItalicText(Object)
- RichText.Text(Object)
- RichText.UnderLineText(Object)
- RichText.StrikeText(Object)
- RichText.SupText(Object)
- RichText.SubText(Object)
- RichText.PositionText(Object, Single)
- RichText.SizeText(Object, Int32)
- RichText.ColoredText(Object, String)
- RichText.ColoredText(Object, Color)
- RichText.BooleanText(Object, Boolean)
- RichText.BooleanText(Object, Boolean, Color)

Class TransferItem

Inheritance

System.Object
TransferItem

Namespace: [InventorySystem.Data](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class TransferItem
```

Constructors

TransferItem(Int32, InventoryItem)

This is the constructor that will create this TransferItem.

Declaration

```
public TransferItem(int key, InventoryItem item)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	key	This is the Key of the item in its owner's dictionary.
InventoryItem	item	This is the Item that is going to be transferred. and it's currentStack will set Amount.

TransferItem(Int32, InventoryItem, Int32)

This is the constructor that will create this TransferItem.

Declaration

```
public TransferItem(int key, InventoryItem item, int amount)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	key	This is the Key of the item in its owner's dictionary.

TYPE	NAME	DESCRIPTION
InventoryItem	item	This is the Item that is going to be transferred.
System.Int32	amount	This is the Amount that is going to be transferred.

Properties

Amount

This is the Amount of the item that will transferred.

Declaration

```
public int Amount { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Cost

This is the Cost of all of the item(s) that will be transferred.

Declaration

```
public int Cost { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Item

This is the Item that is going to be transferred.

Declaration

```
public InventoryItem Item { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryItem	

Key

This is the Key of the item in its owner's dictionary and if the whole item is transferred it will be removed from it's owner.

Declaration

```
public int Key { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Extension Methods

- [RichText.UpperCaseText\(Object\)](#)
- [RichText.LowerCaseText\(Object\)](#)
- [RichText.BoldText\(Object\)](#)
- [RichText.ItalicText\(Object\)](#)
- [RichText.Text\(Object\)](#)
- [RichText.UnderLineText\(Object\)](#)
- [RichText.StrikeText\(Object\)](#)
- [RichText.SupText\(Object\)](#)
- [RichText.SubText\(Object\)](#)
- [RichText.PositionText\(Object, Single\)](#)
- [RichText.SizeText\(Object, Int32\)](#)
- [RichText.ColoredText\(Object, String\)](#)
- [RichText.ColoredText\(Object, Color\)](#)
- [RichText.BooleanText\(Object, Boolean\)](#)
- [RichText.BooleanText\(Object, Boolean, Color\)](#)

Namespace InventorySystem.Editor

Classes

[InventoryWindow](#)

This class is an Editor Window for the creation & editing of the inventory and it's content.

Class InventoryWindow

This class is an Editor Window for the creation & editing of the inventory and it's content.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.ScriptableObject
UnityEditor.EditorWindow
InventoryWindow

Namespace: [InventorySystem.Editor](#)
Assembly: Assembly-CSharp-Editor.dll

Syntax

```
public class InventoryWindow : EditorWindow
```

Properties

SelectedInventory

This is connected to the selectedInventory Variable. If selectedInventory is different from the incoming value then set the value & refresh the UI.

Declaration

```
public Inventory SelectedInventory { get; }
```

Property Value

TYPE	DESCRIPTION
Inventory	

Methods

AddItemData(ItemData)

This will add a new ItemData File to items list Then Add that ItemData to the UI.

Declaration

```
public void AddItemData(ItemData item)
```

Parameters

TYPE	NAME	DESCRIPTION
ItemData	item	This is a new ItemData.

AddItemIntoDictionary(InventoryItem)

This will add the item to the dictionary.

Declaration

```
public bool AddItemIntoDictionary(InventoryItem item)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryItem	item	this is the item that will be added to the dictionary.

Returns

TYPE	DESCRIPTION
System.Boolean	return true if added to dictionary.

ChangeItemData(ItemData)

This gets invoked from an event when the OpenItemUpdate clicks the update or Duplicate button

Declaration

```
public void ChangeItemData(ItemData currentItem)
```

Parameters

TYPE	NAME	DESCRIPTION
ItemData	currentItem	This is the that is changed ItemData.

CloseWindow()

This static will close the InventoryWindow. The window can be open either by double clicking an Inventory file, Though the UnityEditor Toolbar: Tools/DownUnder Studios/Inventory System/Tools/Close Inventory Editor Or by the shortcut key CTRL + SHIFT + ALT + O

Declaration

```
[MenuItem("Tools/DownUnder Studios/Inventory System/Tools/Close Inventory Editor %s")]
public static void CloseWindow()
```

CreateGUI()

This is a unity method, that will be run when the window is created.

Declaration

```
public void CreateGUI()
```

GenerateItemDataFiles()

This will get the ItemData file from the Resources folder and add them to the items list. Then Create UI for each ItemData.

Declaration

```
public void GenerateItemDataFiles()
```

GetInventoryFiles()

This will get all Inventory files in the Resources.

Declaration

```
public void GetInventoryFiles()
```

NewInventory(String)

This will create a new Inventory File And add it to the *inventories* List & Dropdown Field

Declaration

```
public void NewInventory(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	This is the name of the new inventory.

OnOpenAsset(Int32, Int32)

This static will open the InventoryWindow when an Inventory file is double clicked.

Declaration

```
[OnOpenAsset]
public static bool OnOpenAsset(int instanceID, int line)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	instanceID	not used.
System.Int32	line	not used.

Returns

TYPE	DESCRIPTION
System.Boolean	

OpenEditItem(ItemData)

This will open the EditItem Window to edit the selected ItemData.

Declaration

```
public void OpenEditItem(ItemData item)
```

Parameters

TYPE	NAME	DESCRIPTION
ItemData	item	This is the ItemData that will be changed.

OpenWindow()

This static will open the InventoryWindow. The window can be open either by double clicking an Inventory file, Though the UnityEditor Toolbar: Tools/DownUnder Studios/Inventory System/Tools/Open Inventory Editor Or by the shortcut key CTRL + SHIFT + ALT + I

Declaration

```
[MenuItem("Tools/DownUnder Studios/Inventory System/Tools/Open Inventory Editor %#I")]
public static void OpenWindow()
```

Save()

This will Save the inventory's changes. Only if the editor is not in play mode.

Declaration

```
public void Save()
```

SetWindowSize(Int32, Int32)

This will set the Editor Window min & max Size to a fix Size. With both params being the width & height respectively.

Declaration

```
public void SetWindowSize(int width, int height)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	width	The width of the Window.
System.Int32	height	The height of the Window.

UpdateUI()

This will check if SelectedInventory is null or not. If not null set the displays to flex and window Size to 1200-500 else set the displays to none and window Size to 570-30

Declaration

```
public void UpdateUI()
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)

Namespace InventorySystem.Script

Classes

ContextMenuGenerator

This static class will generate Context Menu for inventory.

CreateElements

This static class will allow the creation of UI Toolkit elements.

CustomText

This static class will simplify setting up text for inventory ui.

DemoSceneManager

This is the class that will manage the splashscreen of the demo scene.

DictionaryActions

This static class is used to perform actions on Inventory Dictionary.

Enemy

This class is the script thats applied to a Enemy Character.

EnemyList

EquippedItem

This class should operate as either an additional script on your items, Or a base script for your item so the use method will be invoked by the player script.

GenerateData

This static Class is used to Generate Inventories & InventoryItems from ItemData files.

InventoryManager

This class handles the inventories and logic for transferring of item(s) from 1 inventory to another.

InventorySpawner

This class is the InventorySpawner it will create an inventory scriptable object on awake. and generate its Storage from the list of SpawnItem(s).

InventoryTarget

This class is the base for all classes with with inventory. (Enemy, Player, Storage, Trader)

InventoryTrader

This static Class is used to Trade Inventory item(s).

PickupItem

This class should operate as either an additional script on your items, Or a base script for your item so the use method will be invoked by the player script.

PickupUI

This class will create the pickup ui for each item that is picked up off the ground.

Player

This class is the script thats applied to a player Character.

Storage

This class is the script thats applied to a storage inventory.

Trader

This class is the script thats applied to a Trader Character.

Class ContextMenuGenerator

This static class will generate Context Menu for inventory.

Inheritance

System.Object
ContextMenuGenerator

Namespace: [InventorySystem.Script](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public static class ContextMenuGenerator
```

Methods

ContextMenuButton(InventoryElement, out List<ContextMenuItemData>)

This will create a list of ContextMenuItemData for a ContextMenu.

Declaration

```
public static bool ContextMenuButton(InventoryElement From, out List<ContextMenuItemData> items)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory of the data. This is the inventory of the data.
System.Collections.Generic.List< ContextMenuItemData >	items	The out param is the ContextMenuItemData that was made.

Returns

TYPE	DESCRIPTION
System.Boolean	The return is true.

ContextMenuSlot(InventoryElement, SlotElement, out List<ContextMenuItemData>)

This will create a list of ContextMenuItemData for a ContextMenu.

Declaration

```
public static bool ContextMenuSlot(InventoryElement From, SlotElement slot, out List<ContextMenuItemData> items)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the inventory of the data.
SlotElement	slot	This slot will be the From of the data.
System.Collections.Generic.List< ContextMenuItemData >	items	The out param is the ContextMenuItemData that was made.

Returns

TYPE	DESCRIPTION
System.Boolean	The return is true if items has at list 1 item.

Class CreateElements

This static class will allow the creation of UI Toolkit elements.

Inheritance

System.Object
CreateElements

Namespace: [InventorySystem.Script](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public static class CreateElements
```

Methods

CreateElement_Icon(VisualElement, InventoryItem)

This will create a IconElement and add to the element that invokes this.

Declaration

```
public static IconElement CreateElement_Icon(this VisualElement parent, InventoryItem item)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
InventoryItem	item	This is the item that will be used to create the icon.

Returns

TYPE	DESCRIPTION
IconElement	The return is the IconElement that is created.

CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)

This will create an InventoryElement and add to the element that invokes this.

Declaration

```
public static InventoryElement CreateElement_Inventory(this VisualElement parent, int numberOfSlots, int slotsPerRow, Inventory inventory)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.Int32	numberOfSlots	This is how many slot(s)
System.Int32	slotsPerRow	This how many slot(s) per row.
Inventory	inventory	This is the Inventory to be added to the InventoryElement.

Returns

TYPE	DESCRIPTION
InventoryElement	The return is the InventoryElement that is created.

CreateElement_Slot(VisualElement, Int32, Boolean)

This will create a SlotElement and add to the element that invokes this.

Declaration

```
public static SlotElement CreateElement_Slot(this VisualElement parent, int key, bool isActive)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.Int32	key	This is the key of the slot.
System.Boolean	isActive	

Returns

TYPE	DESCRIPTION
SlotElement	The return is the SlotElement that is created.

CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)

This will create a SlotPopupElement and add to the element that invokes this.

Declaration

```
public static SlotPopupElement CreateElement_SlotPopup(this VisualElement parent, Vector2 size, int key, InventoryElement owner, bool isSplit)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
UnityEngine.Vector2	size	This is the size of the popup.
System.Int32	key	This is the key of the slot.
InventoryElement	owner	This is the InventoryElement that owns the item.
System.Boolean	isSplit	This is the boolean that will determine if it is split or drop.

Returns

TYPE	DESCRIPTION
SlotPopupElement	The return is the SlotPopupElement that is created.

CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

This will create a TransferPopupElement and add to the element that invokes this.

Declaration

```
public static TransferPopupElement CreateElement_TransferPopup(this VisualElement parent, Vector2 size, TransferItem item, InventoryElement To, InventoryElement From)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	size	This is the size of the popup.
TransferItem	item	This is the item that will be transferred.
InventoryElement	To	This is the InventoryElement that owns the item.
InventoryElement	From	This is the InventoryElement that will receive the item.

Returns

TYPE	DESCRIPTION
TransferPopupElement	The return is the TransferPopupElement that is created.

Class CustomText

This static class will simplify setting up text for inventory ui.

Inheritance

System.Object
CustomText

Namespace: [InventorySystem.Script](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public static class CustomText
```

Methods

CurrencyText(Int32)

This will take a 3 int value that represents gold,silver,copper or \$ amount.

Declaration

```
public static string CurrencyText(this int arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	arg	The 3 values to be converted to string.

Returns

TYPE	DESCRIPTION
System.String	This will return a string currency.

WeightText(Single)

This will take a float value that represents weight. And will convert to either metric or imperial. based on weight amount.

Declaration

```
public static string WeightText(this float arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	arg	The weight value to be converted to string.

Returns

TYPE	DESCRIPTION
System.String	This will return a string weight.

Class DemoSceneManager

This is the class that will manage the splashscreen of the demo scene.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
DemoSceneManager

Namespace: [InventorySystem.Script](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class DemoSceneManager : MonoBehaviour
```

Fields

currencyDefault

This button when pressed will set currencyState to true. update currencyExample text.

Declaration

```
public Button currencyDefault
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

currencyExample

This will display an example of when the currency will look like in the ui.

Declaration

```
public Label currencyExample
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Label	

currencyGold

This button when pressed will set currencyState to false. update currencyExample text.

Declaration

```
public Button currencyGold
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

currencyState

This bool is used to update the which currency state should be used in the demo.

Declaration

```
public bool currencyState
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

document

This is the ui document that holds the splash screen.

Declaration

```
public UIDocument document
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.UIDocument	

Instance

This is the singleton of this class.

Declaration

```
public static DemoSceneManager Instance
```

Field Value

TYPE	DESCRIPTION
DemoSceneManager	

startDemo

This button when pressed will start the demo scene.

Declaration

```
public Button startDemo
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

weightExample

This will display an example of when the weight will look like in the ui.

Declaration

```
public Label weightExample
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Label	

weightImperial

This button when pressed will set weightState to false. update currencyExample text.

Declaration

```
public Button weightImperial
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

weightMetric

This button when pressed will set weightState to true. update currencyExample text.

Declaration

```
public Button weightMetric
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

weightState

This bool is used to update the which weight state should be used in the demo.

Declaration

```
public bool weightState
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Extension Methods

- RichText.UpperCaseText(Object)
- RichText.LowerCaseText(Object)
- RichText.BoldText(Object)
- RichText.ItalicText(Object)
- RichText.Text(Object)
- RichText.UnderLineText(Object)
- RichText.StrikeText(Object)
- RichText.SupText(Object)
- RichText.SubText(Object)
- RichText.PositionText(Object, Single)
- RichText.SizeText(Object, Int32)
- RichText.ColoredText(Object, String)
- RichText.ColoredText(Object, Color)
- RichText.BooleanText(Object, Boolean)
- RichText.BooleanText(Object, Boolean, Color)

Class DictionaryActions

This static class is used to perform actions on Inventory Dictionary.

Inheritance

System.Object
DictionaryActions

Namespace: [InventorySystem.Script](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public static class DictionaryActions
```

Methods

AddItemInToDictionary(InventoryElement, Boolean, InventoryItem[])

This will add an item to the dictionary.

Declaration

```
public static bool AddItemInToDictionary(InventoryElement target, bool canMerge, params InventoryItem[] items)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	target	This is the InventoryElement who's dictionary the item will be added to.
System.Boolean	canMerge	The boolean will determine if the item will marge with other dictionary entries of the same item.
InventoryItem[]	items	

Returns

TYPE	DESCRIPTION
System.Boolean	Return true if the item is Added.

AddItemInToDictionary(Dictionary<Int32, InventoryItem>, Int32, Boolean, InventoryItem[])

This will add an item to the dictionary.

Declaration

```
public static bool AddItemInToDictionary(Dictionary<int, InventoryItem> dictionary, int size, bool canMerge,
params InventoryItem[] items)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	dictionary	This is the dictionary that the item will be added to.
System.Int32	size	
System.Boolean	canMerge	The boolean will determine if the item will marge with other dictionary entries of the same item.
InventoryItem []	items	

Returns

TYPE	DESCRIPTION
System.Boolean	Return true if the item is Added.

AmountCanFillOfItem(Dictionary<Int32, InventoryItem>, InventoryItem)

This will check the dictionary for the same item as the param item. and count the free space in each item.

Declaration

```
public static int AmountCanFillofItem(Dictionary<int, InventoryItem> dictionary, InventoryItem item)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	dictionary	This is the dictionary that will be used to get the amount.
InventoryItem	item	This is the Item that will be used to search.

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
System.Int32	The Return is the amount of free space on all of the param item in dictionary.

ConsolidateInventory(Dictionary<Int32, InventoryItem>, Int32)

This will Consolidate InventoryItem(s) into the smallest number of stacks.

Declaration

```
public static void ConsolidateInventory(Dictionary<int, InventoryItem> dictionary, int size)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	dictionary	This is the dictionary that will be consolidated.
System.Int32	size	The size of the dictionary.

ConsolidateInventoryItems(InventoryElement)

This will Consolidate InventoryItem(s) into the smallest number of stacks without moving slots.

Declaration

```
public static void ConsolidateInventoryItems(InventoryElement target)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	target	this is the InventoryElement who's dictionary will be consolidated.

ConvertDictionaryToStorage(Dictionary<Int32, InventoryItem>)

This will add dictionary into the KeyItem list.

Declaration

```
public static List<KeyItem> ConvertDictionaryToStorage(Dictionary<int, InventoryItem> dictionary)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	dictionary	This is the dictionary that will added into the list of KeyItem(s).

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List< KeyItem >	The return is the list of KeyItem(s).

ConvertStorageToDictionary(Dictionary<Int32, InventoryItem>, List<KeyItem>)

This will add KeyItem list into the dictionary.

Declaration

```
public static void ConvertStorageToDictionary(Dictionary<int, InventoryItem> dictionary, List<KeyItem> items)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	dictionary	This is the dictionary that will take the KeyItem(s).
System.Collections.Generic.List< KeyItem >	items	This is the list of KeyItem(s) that will be added into the dictionary.

GetTransferItem(Dictionary<Int32, InventoryItem>, Int32)

This will create a TransferItem for the dictionary value at the param key.

Declaration

```
public static TransferItem GetTransferItem(Dictionary<int, InventoryItem> dictionary, int key)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	dictionary	This is the dictionary that will be used to get the TransferItem's Item.

TYPE	NAME	DESCRIPTION
System.Int32	key	This is the key that will be used to located the item in the dictionary.

Returns

TYPE	DESCRIPTION
TransferItem	The return is the TransferItem that is create.

RemoveItems(InventoryElement, Int32[])

This will remove item(s) from the dictionary.

Declaration

```
public static void RemoveItems(InventoryElement target, params int[] keys)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	target	This is the InventoryElement who's dictionary the item will be removed from.
System.Int32[]	keys	these params will be the keys pointing to the dictionary's entries to be removed.

RemoveItems(Dictionary<Int32, InventoryItem>, Int32[])

This will remove item(s) from the dictionary.

Declaration

```
public static void RemoveItems(Dictionary<int, InventoryItem> dictionary, params int[] keys)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	dictionary	This is the dictionary that the item will be removed from.
System.Int32[]	keys	these params will be the keys pointing to the dictionary's entries to be removed.

SortByAmount(Dictionary<Int32, InventoryItem>, Boolean)

This will sort the dictionary by item current stack, either Ascending or Descending order.

Declaration

```
public static Dictionary<int, InventoryItem> SortByAmount(Dictionary<int, InventoryItem> dictionary, bool isDescending)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	dictionary	This is the dictionary that will be sorted.
System.Boolean	isDescending	This is the boolean that if true will choose to order by Descending.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	Return the dictionary after it has been sorted.

SortByCost(Dictionary<Int32, InventoryItem>, Boolean)

This will sort the dictionary by item cost, either Ascending or Descending order.

Declaration

```
public static Dictionary<int, InventoryItem> SortByCost(Dictionary<int, InventoryItem> dictionary, bool isDescending)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	dictionary	This is the dictionary that will be sorted.
System.Boolean	isDescending	This is the boolean that if true will choose to order by Descending.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	Return the dictionary after it has been sorted.

SortByName(Dictionary<Int32, InventoryItem>, Boolean)

This will sort the dictionary by item name, either Ascending or Descending order.

Declaration

```
public static Dictionary<int, InventoryItem> SortByName(Dictionary<int, InventoryItem> dictionary, bool isDescending)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	dictionary	This is the dictionary that will be sorted.
System.Boolean	isDescending	This is the boolean that if true will choose to order by Descending.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	Return the dictionary after it has been sorted.

SortByType(Dictionary<Int32, InventoryItem>, Boolean)

This will sort the dictionary by item type, either Ascending or Descending order.

Declaration

```
public static Dictionary<int, InventoryItem> SortByType(Dictionary<int, InventoryItem> dictionary, bool isDescending)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	dictionary	This is the dictionary that will be sorted.

TYPE	NAME	DESCRIPTION
System.Boolean	isDescending	This is the boolean that if true will choose to order by Descending.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	Return the dictionary after it has been sorted.

SortByWeight(Dictionary<Int32, InventoryItem>, Boolean)

This will sort the dictionary by item weight, either Ascending or Descending order.

Declaration

```
public static Dictionary<int, InventoryItem> SortByWeight(Dictionary<int, InventoryItem> dictionary, bool isDescending)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	dictionary	This is the dictionary that will be sorted.
System.Boolean	isDescending	This is the boolean that if true will choose to order by Descending.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	Return the dictionary after it has been sorted.

Class Enemy

This class is the script thats applied to a Enemy Character.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
[InventoryTarget](#)
Enemy

Namespace: [InventorySystem.Script](#)

Assembly: Assembly-CSharp.dll

Syntax

```
[RequireComponent(typeof(InventorySpawner))]  
public class Enemy : InventoryTarget
```

Fields

health

This is the heath of the enemy.

Declaration

```
public int health
```

Field Value

TYPE	DESCRIPTION
System.Int32	

onDeath

When invoked will invoke all method subscribed to it.

Declaration

```
public Action onDeath
```

Field Value

TYPE	DESCRIPTION
System.Action	

Properties

Health

This gets health & sets health If health == 0 & isDead == false then set isDead to true & invoke Death Method.

Declaration

```
public int Health { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

isDead

This if true will stop Death Action from being invoke.

Declaration

```
public bool isDead { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

Start()

This is a unity method, will set the decay timer. and subscribe the 'decay.StartTimer' and 'InitInventory' methods to the onDeath action

Declaration

```
public void Start()
```

Update()

This is a unity method, and will UpdateTimer on the decay timer unless the inventory is open or the decay timer hasn't started yet.

Declaration

```
public void Update()
```

Extension Methods

`RichText.UpperCaseText(Object)`

`RichText.LowerCaseText(Object)`

`RichText.BoldText(Object)`

`RichText.ItalicText(Object)`

`RichText.Text(Object)`

`RichText.UnderLineText(Object)`

`RichText.StrikeText(Object)`

`RichText.SupText(Object)`

`RichText.SubText(Object)`

`RichText.PositionText(Object, Single)`

`RichText.SizeText(Object, Int32)`

`RichText.ColoredText(Object, String)`

`RichText.ColoredText(Object, Color)`

`RichText.BooleanText(Object, Boolean)`

`RichText.BooleanText(Object, Boolean, Color)`

Class EnemyList

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
EnemyList

Namespace: [InventorySystem.Script](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class EnemyList : MonoBehaviour
```

Fields

enemies

This is the list of enemy that will be killed.

Declaration

```
public List<Enemy> enemies
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< Enemy >	

Methods

Awake()

This will get all of the enemy types in child gameobject.

Declaration

```
public void Awake()
```

KillEnemies()

This will kill all of the enemy inventories.

Declaration

```
public void KillEnemies()
```

Update()

This will check inputs to determine to kill enemies and open-close inventories.

Declaration

```
public void Update()
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)

Class EquippedItem

This class should operate as either an additional script on your items, Or a base script for your item so the use method will be invoked by the player script.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
[WorldItem](#)
EquippedItem

Namespace: [InventorySystem.Script](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class EquippedItem : WorldItem
```

Methods

Use()

This will be invoked when the player invokes "useltem" action. This will use the item wether as a weapon or other.

Declaration

```
public virtual void Use()
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)
[RichText.LowerCaseText\(Object\)](#)
[RichText.BoldText\(Object\)](#)
[RichText.ItalicText\(Object\)](#)
[RichText.Text\(Object\)](#)
[RichText.UnderLineText\(Object\)](#)
[RichText.StrikeText\(Object\)](#)
[RichText.SupText\(Object\)](#)
[RichText.SubText\(Object\)](#)
[RichText.PositionText\(Object, Single\)](#)
[RichText.SizeText\(Object, Int32\)](#)
[RichText.ColoredText\(Object, String\)](#)
[RichText.ColoredText\(Object, Color\)](#)
[RichText.BooleanText\(Object, Boolean\)](#)
[RichText.BooleanText\(Object, Boolean, Color\)](#)

Class GenerateData

This static Class is used to Generate Inventories & InventoryItems from ItemData files.

Inheritance

System.Object
GenerateData

Namespace: [InventorySystem.Script](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public static class GenerateData
```

Properties

NewInventory

This Property will create an instance of the scriptableObject 'Inventory'.

Declaration

```
public static Inventory NewInventory { get; }
```

Property Value

TYPE	DESCRIPTION
Inventory	The return is the Inventory that was created.

Methods

CreateInventoryItem(String, Int32)

This will create an InventoryItem and add an amount to it.

Declaration

```
public static InventoryItem CreateInventoryItem(string targetIID, int amount = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	targetIID	the Item IID.

TYPE	NAME	DESCRIPTION
System.Int32	amount	the amount to be added to the InventoryItem.

Returns

TYPE	DESCRIPTION
InventoryItem	Return the InventoryItem that was created.

CreateRandomInventoryItem()

This will create a random InventoryItem and add a random amount to it.

Declaration

```
public static InventoryItem CreateRandomInventoryItem()
```

Returns

TYPE	DESCRIPTION
InventoryItem	Return the InventoryItem that was created.

FindItemData(String)

This will to located the ItemData by IID.

Declaration

```
public static ItemData FindItemData(string targetIID)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	targetIID	the IID of the ItemData to be found.

Returns

TYPE	DESCRIPTION
ItemData	Return ItemData file after it is found.

SetItems()

This will get all ItemData from the Resources folder and add them to the items list.

Declaration

```
public static void SetItems()
```

TryFindItemData(String, out ItemData)

This will try to located the ItemData by IID.

Declaration

```
public static bool TryFindItemData(string targetIID, out ItemData itemData)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	targetIID	the IID of the ItemData to be found.
ItemData	itemData	out ItemData file after it is found.

Returns

TYPE	DESCRIPTION
System.Boolean	Return true if ItemData file is found.

Class InventoryManager

This class handles the inventories and logic for transferring of item(s) from 1 inventory to another.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
InventoryManager

Namespace: [InventorySystem.Script](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[RequireComponent(typeof(UIDocument))]  
public class InventoryManager : MonoBehaviour
```

Fields

contextMenu

Declaration

```
public ContextMenuElement contextMenu
```

Field Value

TYPE	DESCRIPTION
ContextMenuElement	

CurrencyState

This is the value that is used to determine what currency will be used.

Declaration

```
[SerializeField]  
public bool CurrencyState
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Instance

This is a singleton Instance variable.

Declaration

```
public static InventoryManager Instance
```

Field Value

TYPE	DESCRIPTION
InventoryManager	

WeightState

This is the value that is used to determine what weight will be used.

Declaration

```
[SerializeField]  
public bool WeightState
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

Properties

CountAll

This Property is the count of the inventories.

Declaration

```
public int CountAll { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

CountLoots

This Property is the count of the inventories of type Loot.

Declaration

```
public int CountLoots { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

CountPlayers

This Property is the count of the inventories of type Player.

Declaration

```
public int CountPlayers { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

CountStorages

This Property is the count of the inventories of type Storage.

Declaration

```
public int CountStorages { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

CountTraders

This Property is the count of the inventories of type Trader.

Declaration

```
public int CountTraders { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

dialogBoxes

This is the list of DialogBoxes that hold an InventoryElement.

Declaration

```
public List<DialogBoxElement> dialogBoxes { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List< DialogBoxElement >	

FindAll

This Property is all of the InventoryElements in inventories.

Declaration

```
public InventoryElement[] FindAll { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryElement []	

FindLoots

This Property is all of the InventoryElements in inventories of type Loot.

Declaration

```
public InventoryElement[] FindLoots { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryElement []	

FindPlayer

This Property is all of the InventoryElements in inventories of type Player.

Declaration

```
public InventoryElement FindPlayer { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryElement	

FindStorages

This Property is all of the InventoryElements in inventories of type Storage.

Declaration

```
public InventoryElement[] FindStorages { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryElement[]	

FindTraders

This Property is all of the InventoryElements in inventories of type Trader.

Declaration

```
public InventoryElement[] FindTraders { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryElement[]	

Methods

AddInventory(InventoryElement)

This will add an InventoryElement to the inventories list.

Declaration

```
public void AddInventory(InventoryElement inventory)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	inventory	This is the inventory that will added.

ContextMenu(InventoryElement, SlotElement)

This will Create a context menu from the button on the inventory.

Declaration

```
public void ContextMenu(InventoryElement target, SlotElement slot)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	target	This is the target InventoryElement that will be used to get the Context Menu Item Data from.
SlotElement	slot	This is the SlotElement that will be used to set the position of the context menu on the screen.

ContextMenu(InventoryElement, Button)

This will Create a context menu from the button on the inventory.

Declaration

```
public void ContextMenu(InventoryElement target, Button button)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	target	This is the target InventoryElement that will be used to get the Context Menu Item Data from.
UnityEngine.UIElements.Button	button	This is the button that will be used to set the position of the context menu on the screen.

CreateInventoryElement(Inventory, Int32, Int32, Vector2, out DialogBoxElement)

This will create the InventoryElement for the inventory.

Declaration

```
public bool CreateInventoryElement(Inventory inventory, int size, int row, Vector2 position, out DialogBoxElement dialogBox)
```

Parameters

TYPE	NAME	DESCRIPTION
Inventory	inventory	This is the inventory that requires a UI for interaction.
System.Int32	size	This is the number of Slots that the UI will have.
System.Int32	row	
UnityEngine.Vector2	position	This is the position of the DialogBox UI will be set at.
DialogBoxElement	dialogBox	This out is the DialogBoxElement so it can be returned for the created script.

Returns

TYPE	DESCRIPTION
System.Boolean	This will return true if the dialogbox is created for the UI.

DetailBox(Int32, InventoryElement)

This will create a DialogBoxElement for the item details.

Declaration

```
public void DetailBox(int key, InventoryElement inventoryElement)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	key	This is the key for the dictionary's key.
InventoryElement	inventoryElement	This param is the target that holds the dictionary.

GetCountTypes(InventoryElement)

Check how many inventory type are in the inventories

Declaration

```
public int GetCountTypes(InventoryElement target)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	target	

Returns

TYPE	DESCRIPTION
System.Int32	

OpenSlotPopup(Int32, InventoryElement, Boolean)

Declaration

```
public void OpenSlotPopup(int key, InventoryElement From, bool isSplit)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	key	
InventoryElement	From	
System.Boolean	isSplit	

OpenTransferPopup(TransferItem, InventoryElement, InventoryElement)

This will open a popup UI for transfer an item from an inventory to another.

Declaration

```
public void OpenTransferPopup(TransferItem item, InventoryElement To, InventoryElement From)
```

Parameters

TYPE	NAME	DESCRIPTION
TransferItem	item	This is the TransferItem that will be the target of transfer.

TYPE	NAME	DESCRIPTION
InventoryElement	To	This is the inventory that will own that item when traded.
InventoryElement	From	This is the owner of the item.

RemoveContextMenu()

This will remove the context menu from the UI if it exist.

Declaration

```
public void RemoveContextMenu()
```

RemoveInventory(InventoryElement)

This will remove an InventoryElement to the inventories list.

Declaration

```
public void RemoveInventory(InventoryElement inventory)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryElement	inventory	This is the inventory that will removed.

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)

Class InventorySpawner

This class is the InventorySpawner it will create an inventory scriptable object on awake. and generate its Storage from the list of SpawnItem(s).

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
InventorySpawner

Namespace: [InventorySystem.Script](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class InventorySpawner : MonoBehaviour
```

Properties

Inventory

This is the Inventory that will be create.

Declaration

```
[SerializeField]  
public Inventory Inventory { get; }
```

Property Value

TYPE	DESCRIPTION
Inventory	

Methods

GenerateInventoryStorage()

This will create a InventoryItem(s) from the spawnList and add them to a list of KeyItem(s).

Declaration

```
public List<KeyItem> GenerateInventoryStorage()
```

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List< KeyItem >	The return is the list of KeyItem(s).

Extension Methods

- [RichText.UpperCaseText\(Object\)](#)
- [RichText.LowerCaseText\(Object\)](#)
- [RichText.BoldText\(Object\)](#)
- [RichText.ItalicText\(Object\)](#)
- [RichText.Text\(Object\)](#)
- [RichText.UnderLineText\(Object\)](#)
- [RichText.StrikeText\(Object\)](#)
- [RichText.SupText\(Object\)](#)
- [RichText.SubText\(Object\)](#)
- [RichText.PositionText\(Object, Single\)](#)
- [RichText.SizeText\(Object, Int32\)](#)
- [RichText.ColoredText\(Object, String\)](#)
- [RichText.ColoredText\(Object, Color\)](#)
- [RichText.BooleanText\(Object, Boolean\)](#)
- [RichText.BooleanText\(Object, Boolean, Color\)](#)

Class InventoryTarget

This class is the base for all classes with with inventory. (Enemy, Player, Storage, Trader)

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
InventoryTarget

[Enemy](#)
[Player](#)
[Storage](#)
[Trader](#)

Namespace: [InventorySystem.Script](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class InventoryTarget : MonoBehaviour
```

Fields

inventory

This is the character's inventory.

Declaration

```
[SerializeField]  
protected Inventory inventory
```

Field Value

TYPE	DESCRIPTION
Inventory	

inventoryRowLength

This is how many slots will be on each row of the inventoryUI;

Declaration

```
[SerializeField]  
protected int inventoryRowLength
```

Field Value

TYPE	DESCRIPTION
System.Int32	

inventorySlotLength

This is how many slots will be on the inventoryUI;

Declaration

```
[SerializeField]
protected int inventorySlotLength
```

Field Value

TYPE	DESCRIPTION
System.Int32	

inventoryUI

This is the DialogBox that contains the inventory UI.

Declaration

```
protected DialogBoxElement inventoryUI
```

Field Value

TYPE	DESCRIPTION
DialogBoxElement	

inventoryUIPosition

This will be that value used to set the position of the UI on the screen.

Declaration

```
[SerializeField]
protected Vector2 inventoryUIPosition
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Vector2	

Properties

GetContent

This is the InventoryElement for this character.

Declaration

```
public InventoryElement GetContent { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryElement	

HasInventory

This is a property check that will see if 'inventoryUI' isn't null.

Declaration

```
public bool HasInventory { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

IsInventoryOpen

Declaration

```
public bool IsInventoryOpen { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

CloseInventory()

This will close the inventoryUI

Declaration

```
public void CloseInventory()
```

InitInventory()

This will initialise the inventoryUI for this character.

Declaration

```
public void InitInventory()
```

OnDestroy()

This is a unity method, and will remove the UI from its parent element.

Declaration

```
public void OnDestroy()
```

OpenInventory()

This will open the inventory

Declaration

```
public void OpenInventory()
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)

Class InventoryTrader

This static Class is used to Trade Inventory item(s).

Inheritance

System.Object
InventoryTrader

Namespace: [InventorySystem.Script](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public static class InventoryTrader
```

Methods

Cost(TransferItem[])

This will get the cost of the purchase.

Declaration

```
public static int Cost(params TransferItem[] transferItems)
```

Parameters

TYPE	NAME	DESCRIPTION
TransferItem[]	transferItems	params of TransferItem(s) that have their cost added together.

Returns

TYPE	DESCRIPTION
System.Int32	return the cost of the purchase.

ItemsToBuy((Inventory, Dictionary<Int32, InventoryItem>, Action), TransferItem[])

This will check a list of TransferItem(s) how many of each item will be purchase.

Declaration

```
public static List<TransferItem> ItemsToBuy((Inventory, Dictionary<int, InventoryItem>, Action) To, params TransferItem[] items)
```

Parameters

TYPE	NAME	DESCRIPTION
System.ValueTuple< Inventory , System.Collections.Generic.Dictionary<System.Int32, InventoryItem >, System.Action>	To	The Inventory that will be the purchaser of the items.
TransferItem []	items	The list of the TransferItem that will be purchased.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List< TransferItem >	This will return the list of TransferItem(s).

ItemsToTransfer((Inventory, Dictionary<Int32, InventoryItem>, Action), TransferItem[])

This will check if their is enough free slots for the amount of items.

Declaration

```
public static List<TransferItem> ItemsToTransfer((Inventory, Dictionary<int, InventoryItem>, Action) To,
params TransferItem[] items)
```

Parameters

TYPE	NAME	DESCRIPTION
System.ValueTuple< Inventory , System.Collections.Generic.Dictionary<System.Int32, InventoryItem >, System.Action>	To	The Inventory that will be the purchaser of the item.
TransferItem []	items	The list of the TransferItem that will be Transferred.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List< TransferItem >	This will return the list of TransferItem(s).

ItemToPurchase(TransferItem, Int32)

Declaration

```
public static (TransferItem, int) ItemToPurchase(TransferItem item, int currencyValue)
```

Parameters

TYPE	NAME	DESCRIPTION
TransferItem	item	
System.Int32	currencyValue	

Returns

TYPE	DESCRIPTION
System.ValueTuple< TransferItem , System.Int32>	

TransferItems((Inventory, Dictionary<Int32, InventoryItem>, Action), (Inventory, Dictionary<Int32, InventoryItem>, Action), TransferItem[])

This will transfer all of the TransferItem(s).

Declaration

```
public static void TransferItems((Inventory, Dictionary<int, InventoryItem>, Action) From, (Inventory, Dictionary<int, InventoryItem>, Action) To, params TransferItem[] items)
```

Parameters

TYPE	NAME	DESCRIPTION
System.ValueTuple< Inventory , System.Collections.Generic.Dictionary<System.Int32, InventoryItem >, System.Action>	From	The Inventory that will be the Seller of the item.
System.ValueTuple< Inventory , System.Collections.Generic.Dictionary<System.Int32, InventoryItem >, System.Action>	To	The Inventory that will be the purchaser of the item.
TransferItem []	items	These TransferItem(s) that will be traded.

TransferMoney(Inventory, Inventory)

This will transfer all of the currency From an inventory To another.

Declaration

```
public static void TransferMoney(Inventory From, Inventory To)
```

Parameters

TYPE	NAME	DESCRIPTION
Inventory	From	The Inventory that will be the Seller of the item.
Inventory	To	The Inventory that will be the purchaser of the item.

TransferMoney(Inventory, Inventory, Int32)

This will transfer an amount of currency From an inventory To another.

Declaration

```
public static void TransferMoney(Inventory To, Inventory From, int amount)
```

Parameters

TYPE	NAME	DESCRIPTION
Inventory	To	The Inventory that will be the Seller of the item.
Inventory	From	The Inventory that will be the purchaser of the item.
System.Int32	amount	The amount of currency that will be traded.

Class PickupItem

This class should operate as either an additional script on your items, Or a base script for your item so the use method will be invoked by the player script.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour

[WorldItem](#)

PickupItem

Namespace: [InventorySystem.Script](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class PickupItem : WorldItem
```

Methods

PickUp()

This will be invoked when the player invokes "useltem" action. This will pickup the item and add it to the player's inventory.

Declaration

```
public void PickUp()
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)
[RichText.LowerCaseText\(Object\)](#)
[RichText.BoldText\(Object\)](#)
[RichText.ItalicText\(Object\)](#)
[RichText.Text\(Object\)](#)
[RichText.UnderLineText\(Object\)](#)
[RichText.StrikeText\(Object\)](#)
[RichText.SupText\(Object\)](#)
[RichText.SubText\(Object\)](#)
[RichText.PositionText\(Object, Single\)](#)
[RichText.SizeText\(Object, Int32\)](#)
[RichText.ColoredText\(Object, String\)](#)
[RichText.ColoredText\(Object, Color\)](#)
[RichText.BooleanText\(Object, Boolean\)](#)
[RichText.BooleanText\(Object, Boolean, Color\)](#)

Class PickupUI

This class will create the pickup ui for each item that is picked up off the ground.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
PickupUI

Namespace: [InventorySystem.Script](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[RequireComponent(typeof(UIDocument))]  
public class PickupUI : MonoBehaviour
```

Fields

document

This is the UIDocument that will hold the pickup item pop up.

Declaration

```
public UIDocument document
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.UIDocument	

Instance

This is the singleton of this class.

Declaration

```
public static PickupUI Instance
```

Field Value

TYPE	DESCRIPTION
PickupUI	

pickupParent

This is the parent of the pickups.

Declaration

```
public VisualElement pickupParent
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.VisualElement	

Methods

Awake()

This will set the pickupParent from the document.

Declaration

```
public void Awake()
```

PickupItem(ItemData, Int32)

This will run after an item has been picked up and added to inventory.

Declaration

```
public void PickupItem(ItemData item, int amount)
```

Parameters

TYPE	NAME	DESCRIPTION
ItemData	item	the Item that was picked up.
System.Int32	amount	

Update()

This will update all of the timers.

Declaration

```
public void Update()
```

Extension Methods

`RichText.UpperCaseText(Object)`

`RichText.LowerCaseText(Object)`

`RichText.BoldText(Object)`

`RichText.ItalicText(Object)`

`RichText.Text(Object)`

`RichText.UnderLineText(Object)`

`RichText.StrikeText(Object)`

`RichText.SupText(Object)`

`RichText.SubText(Object)`

`RichText.PositionText(Object, Single)`

`RichText.SizeText(Object, Int32)`

`RichText.ColoredText(Object, String)`

`RichText.ColoredText(Object, Color)`

`RichText.BooleanText(Object, Boolean)`

`RichText.BooleanText(Object, Boolean, Color)`

Class Player

This class is the script thats applied to a player Character.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
[InventoryTarget](#)
Player

Namespace: [InventorySystem.Script](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class Player : InventoryTarget
```

Fields

equippedItem

This is the gameobject for the item in the playerHand.

Declaration

```
public GameObject equippedItem
```

Field Value

TYPE	DESCRIPTION
UnityEngine.GameObject	

Instance

This is a singleton instance.

Declaration

```
public static Player Instance
```

Field Value

TYPE	DESCRIPTION
Player	

playerHand

This is the gameobject for the hand. When item is equipped it will be apply to the hand.

Declaration

```
public GameObject playerHand
```

Field Value

TYPE	DESCRIPTION
UnityEngine.GameObject	

useItem

This is for when the equippedItem is used.

Declaration

```
public Action useItem
```

Field Value

TYPE	DESCRIPTION
System.Action	

Methods

DestroyHeldObjects()

This will destroy the items in the player's hand.

Declaration

```
public void DestroyHeldObjects()
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Class Storage

This class is the script thats applied to a storage inventory.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
[InventoryTarget](#)
Storage

Namespace: [InventorySystem.Script](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class Storage : InventoryTarget
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)
[RichText.LowerCaseText\(Object\)](#)
[RichText.BoldText\(Object\)](#)
[RichText.ItalicText\(Object\)](#)
[RichText.Text\(Object\)](#)
[RichText.UnderLineText\(Object\)](#)
[RichText.StrikeText\(Object\)](#)
[RichText.SupText\(Object\)](#)
[RichText.SubText\(Object\)](#)
[RichText.PositionText\(Object, Single\)](#)
[RichText.SizeText\(Object, Int32\)](#)
[RichText.ColoredText\(Object, String\)](#)
[RichText.ColoredText\(Object, Color\)](#)
[RichText.BooleanText\(Object, Boolean\)](#)
[RichText.BooleanText\(Object, Boolean, Color\)](#)

Class Trader

This class is the script thats applied to a Trader Character.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
[InventoryTarget](#)
Trader

Namespace: [InventorySystem.Script](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class Trader : InventoryTarget
```

Fields

currency

This is the currency of the trader when it gets a restock.

Declaration

```
public int currency
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Methods

Restock()

This is the method that will be called for restocking of the inventory. it will also reset the timer so when it the inventoryUI is open again it will start the timer again.

Declaration

```
public void Restock()
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Namespace InventorySystem.UI

Classes

[IconElement](#)

This class is an element that is created for the Inventory.

[IconElement.UxmlFactory](#)

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

[InventoryElement](#)

this class is an element for the inventory.

[InventoryElement.UxmlFactory](#)

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

[SlotElement](#)

This class is an element that is created for the Inventory.

[SlotElement.UxmlFactory](#)

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

[SlotPopupElement](#)

This class is the popup for the Slot, for the splitting and dropping.

[SlotPopupElement.UxmlFactory](#)

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

[TransferPopupElement](#)

This class is the popup for the Slot, for the selling and trading.

[TransferPopupElement.UxmlFactory](#)

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Class IconElement

This class is an element that is created for the Inventory.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
IconElement

Namespace: [InventorySystem.UI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class IconElement : VisualElement, IEventHandler, ITransform, ITransitionAnimations,  
IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

IconElement()

This is a constructor and will set the style for the Icon.

Declaration

```
public IconElement()
```

IconElement(InventoryItem, StyleSheet)

Declaration

```
public IconElement(InventoryItem item, StyleSheet style = null)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryItem	item	
UnityEngine.UIElements.StyleSheet	style	

Methods

ItemAmount(Int32)

This will create a Label for the amount that the item currently has. Set the style of the Label and adds to the Icon.

Declaration

```
public void ItemAmount(int amount)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	amount	the amount that the item currently has

Select()

This will set the border color to green to show the user it is selected.

Declaration

```
public void Select()
```

Style()

This will set the style of the Icon.

Declaration

```
public void Style()
```

UnSelect()

This will set the border color to auto to show the user it is unselected.

Declaration

```
public void UnSelect()
```

UpdateIcon(Int32)

This will update the Icon with a new amount.

Declaration

```
public bool UpdateIcon(int amount)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	amount	The amount to update item amount.

Returns

TYPE	DESCRIPTION
System.Boolean	Return true when updated.

Extension Methods

VisualElementStyleExtension.BackGround_Color(VisualElement, Color)
VisualElementStyleExtension.Background_Image(VisualElement, Background)
VisualElementStyleExtension.Background_Image(VisualElement, Texture2D)
VisualElementStyleExtension.Background_Image(VisualElement, Sprite)
VisualElementStyleExtension.Background_Image(VisualElement, StyleKeyword)
VisualElementStyleExtension.Color(VisualElement, Color)
VisualElementStyleExtension.Color(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Top(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Right(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Left(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color(VisualElement, Color[])
VisualElementStyleExtension.Border_Color(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetRadius_Top(VisualElement)
VisualElementStyleExtension.GetRadius_Bottom(VisualElement)
VisualElementStyleExtension.GetRadius_Size(VisualElement)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius(VisualElement, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetBorder_Width(VisualElement)
VisualElementStyleExtension.GetBorder_Height(VisualElement)
VisualElementStyleExtension.GetBorder_Size(VisualElement)
VisualElementStyleExtension.Border_Width_Top(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Right(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Left(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Left(VisualElement, StyleKeyword)

VisualElementStyleExtension.Border_Width(VisualElement, Single[])
VisualElementStyleExtension.Border_Width(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Border(VisualElement, Single[])
VisualElementStyleExtension.Border(VisualElement, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)
VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)

VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)

VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)
VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)
VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)
VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)
VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)
VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)
VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)

VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)
VisualElementExtension.SetTransform_Position(VisualElement, Vector2)
VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)
VisualElementExtension.GetElementSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)
VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)
CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)
CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)

CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32, Boolean)
CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)
CreateElements.CreateElement_Icon(VisualElement, InventoryItem)
CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)
CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class IconElement.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Inheritance

System.Object

UnityEngine.UIElements.UxmlFactory<[IconElement](#), [VisualElement.UxmlTraits](#)>

IconElement.UxmlFactory

Namespace: [InventorySystem.UI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<IconElement, VisualElement.UxmlTraits>, IUxmlFactory
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)

Class InventoryElement

this class is an element for the inventory.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
InventoryElement

Namespace: [InventorySystem.UI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class InventoryElement : VisualElement, IEventHandler, ITransform, ITransitionAnimations,
    IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

InventoryElement()

This is a constructor and will set the style for the Slot.

Declaration

```
public InventoryElement()
```

InventoryElement(Int32, Int32, Inventory, StyleSheet)

This is the constructor for this class.

Declaration

```
public InventoryElement(int numberOfSlots, int slotsPerRow, Inventory Inventory, StyleSheet style = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	numberOfSlots	This is the number of slots that will be added to the UI.
System.Int32	slotsPerRow	
Inventory	Inventory	This is the Inventory for this UI.

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.StyleSheet	style	This is an optional StyleSheet.

Fields

currentSortOrder

This enum is used for sorting the UI item.

Declaration

```
public SortByEnum currentSortOrder
```

Field Value

TYPE	DESCRIPTION
SortByEnum	

dictionary

This is the Dictionary that will store the InventoryItem(s).

Declaration

```
public Dictionary<int, InventoryItem> dictionary
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.Int32, InventoryItem >	

header

This is the header section.

Declaration

```
public VisualElement header
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.VisualElement	

isDescending

this is used in sort and if it should be ascending or descending.

Declaration

```
public bool isDescending
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

numberOfSlots

This is the number of slots in the ui.

Declaration

```
public int numberOfSlots
```

Field Value

TYPE	DESCRIPTION
System.Int32	

onUpdate

This is the Action that will be called when the ui will be updated.

Declaration

```
public Action onUpdate
```

Field Value

TYPE	DESCRIPTION
System.Action	

slotParent

This is the parent element that hold the slot(s).

Declaration

```
public VisualElement slotParent
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.VisualElement	

slots

This is the list of slots on the UI.

Declaration

```
[SerializeField]  
public List<SlotElement> slots
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< SlotElement >	

slotsToUpdate

This is a list of slots that need to be update.

Declaration

```
public List<int> slotsToUpdate
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List<System.Int32>	

titleName

This is the string that will be used on the UI title.

Declaration

```
public string titleName
```

Field Value

TYPE	DESCRIPTION
System.String	

Properties

AllTransferItems

This will return a list of all TransferItem(s) from the dictionary.

Declaration

```
public List<TransferItem> AllTransferItems { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List< TransferItem >	

ArmourTransferItems

This will return a list of TransferItem(s) from the dictionary of type Armour.

Declaration

```
public List<TransferItem> ArmourTransferItems { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List< TransferItem >	

ConsumableTransferItems

This will return a list of TransferItem(s) from the dictionary of type Consumable.

Declaration

```
public List<TransferItem> ConsumableTransferItems { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List< TransferItem >	

Inventory

This property is the Inventory of the character.

Declaration

```
public Inventory Inventory { get; }
```

Property Value

TYPE	DESCRIPTION
Inventory	

MaterialsTransferItems

This will return a list of TransferItem(s) from the dictionary of type Materials.

Declaration

```
public List<TransferItem> MaterialsTransferItems { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List< TransferItem >	

MiscTransferItems

This will return a list of TransferItem(s) from the dictionary of type Misc.

Declaration

```
public List<TransferItem> MiscTransferItems { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List< TransferItem >	

Type

This property is the Inventory's Type.

Declaration

```
[SerializeField]  
public InventoryType Type { get; }
```

Property Value

TYPE	DESCRIPTION
InventoryType	

UID

This property is the Unique id from the Inventory.

Declaration

```
public string UID { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

WeaponTransferItems

This will return a list of TransferItem(s) from the dictionary of type Weapon.

Declaration

```
public List<TransferItem> WeaponTransferItems { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.List< TransferItem >	

Methods

GenerateSlots()

This will create and set up the slotElements.

Declaration

```
public void GenerateSlots()
```

OnClose()

This should be invoked when closing the UI.

Declaration

```
public void OnClose()
```

OnOpen()

This will be invoked in constructor.

Declaration

```
public void OnOpen()
```

Remarks

should be invoked when opening UI. (if the onClose is used.)

SetCurrencyValue(Int32, Int32)

This will set the currencyValue from when the inventory's currency is changed.

Declaration

```
public void SetCurrencyValue(int gold, int valueChanged)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	gold	This is the Currency Value.
System.Int32	valueChanged	This is how much it is changed by.

SetWeightValue()

This will set the weightValue from when the inventory's updated.

Declaration

```
public void SetWeightValue()
```

Sortby(SortByEnum)

This will sort the dictionary by enum.

Declaration

```
public void Sortby(SortByEnum sortBy)
```

Parameters

TYPE	NAME	DESCRIPTION
SortByEnum	sortBy	This param is an enum that will be used to choose what the dictionary will be sorted by.

ToggleInteractable(Boolean)

This will allow or disallow the user to interact with any of the slots.

Declaration

```
public void ToggleInteractable(bool value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	value	This value will determine if interactions will be allowed or disallowed.

UpdateAllSlots()

This will update all of the slots with corresponding dictionary item or null.

Declaration

```
public void UpdateAllSlots()
```

UpdateSlots()

This will update only the slot(s) that were added to the slotsToUpdate list. then clear the slotsToUpdate list.

Declaration

```
public void UpdateSlots()
```

Extension Methods

[VisualElementStyleExtension.BackgroundColor\(VisualElement, Color\)](#)

[VisualElementStyleExtension.BackgroundImage\(VisualElement, Background\)](#)

[VisualElementStyleExtension.BackgroundImage\(VisualElement, Texture2D\)](#)

VisualElementStyleExtension.Background_Image(VisualElement, Sprite)
VisualElementStyleExtension.Background_Image(VisualElement, StyleKeyword)
VisualElementStyleExtension.Color(VisualElement, Color)
VisualElementStyleExtension.Color(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Top(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Right(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Left(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color(VisualElement, Color[])
VisualElementStyleExtension.Border_Color(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetRadius_Top(VisualElement)
VisualElementStyleExtension.GetRadius_Bottom(VisualElement)
VisualElementStyleExtension.GetRadius_Size(VisualElement)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius(VisualElement, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetBorder_Width(VisualElement)
VisualElementStyleExtension.GetBorder_Height(VisualElement)
VisualElementStyleExtension.GetBorder_Size(VisualElement)
VisualElementStyleExtension.Border_Width_Top(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Right(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Left(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width(VisualElement, Single[])
VisualElementStyleExtension.Border_Width(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Border(VisualElement, Single[])
VisualElementStyleExtension.Border(VisualElement, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)
VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)

VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])

VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)
VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)
VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)

VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)
VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)
VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)
VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)
VisualElementExtension.SetTransform_Position(VisualElement, Vector2)
VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)
VisualElementExtension.GetElementSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)

VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)
CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)
CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32, Boolean)
CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)
CreateElements.CreateElement_Icon(VisualElement, InventoryItem)
CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)
CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class InventoryElement.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Inheritance

System.Object

UnityEngine.UIElements.UxmlFactory<[InventoryElement](#), [VisualElement.UxmlTraits](#)>

InventoryElement.UxmlFactory

Namespace: [InventorySystem.UI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<InventoryElement, VisualElement.UxmlTraits>, IUxmlFactory
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)

Class SlotElement

This class is an element that is created for the Inventory.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
SlotElement

Namespace: [InventorySystem.UI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class SlotElement : VisualElement, IEventHandler, ITransform, ITransitionAnimations,
    IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

SlotElement()

This is a constructor and will set the style for the Slot.

Declaration

```
public SlotElement()
```

SlotElement(Int32, Boolean, StyleSheet)

This is a constructor and will set the style for the Slot.

Declaration

```
public SlotElement(int key, bool isActive = true, StyleSheet style = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	key	this is the slots position in the ui list.
System.Boolean	isActive	this will determine if the slot can be interacted with.
UnityEngine.UIElements.StyleSheet	style	This is if you want ot add a stylesheet to this element.

Properties

Active

Declaration

```
public bool Active { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

GetIcon

This property will get Icon attached to this Slot.

Declaration

```
[SerializeField]  
public IconElement GetIcon { get; }
```

Property Value

TYPE	DESCRIPTION
IconElement	Return the Icon.

HasIcon

This property will check if the slot has an icon.

Declaration

```
[SerializeField]  
public bool HasIcon { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Key

This is the key that corresponds to the dictionary entry's key.

Declaration

```
public int Key { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Methods

ClearIcon()

This will remove the icon from this slot.

Declaration

```
public void ClearIcon()
```

SetIcon(InventoryItem)

This will create a new Icon to this slot using an InventoryItem.

Declaration

```
public void SetIcon(InventoryItem item)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryItem	item	The InventoryItem that will be used to on the Icon.

SetIcon(IconElement)

This will create a new Icon to this slot using an InventoryItem.

Declaration

```
public void SetIcon(IconElement icon)
```

Parameters

TYPE	NAME	DESCRIPTION
IconElement	icon	

Style()

This will set the style of the Icon.

Declaration

```
public virtual void Style()
```

UpdateSlot(InventoryItem, Boolean)

This will update the slot's icon with the new item.

Declaration

```
public void UpdateSlot(InventoryItem item, bool isActive)
```

Parameters

TYPE	NAME	DESCRIPTION
InventoryItem	item	this is the item that will update the slot's icon.
System.Boolean	isActive	

Extension Methods

- [VisualElementStyleExtension.BackgroundColor\(VisualElement, Color\)](#)
- [VisualElementStyleExtension.BackgroundImage\(VisualElement, Background\)](#)
- [VisualElementStyleExtension.BackgroundImage\(VisualElement, Texture2D\)](#)
- [VisualElementStyleExtension.BackgroundImage\(VisualElement, Sprite\)](#)
- [VisualElementStyleExtension.BackgroundImage\(VisualElement, StyleKeyword\)](#)
- [VisualElementStyleExtension.Color\(VisualElement, Color\)](#)
- [VisualElementStyleExtension.Color\(VisualElement, StyleKeyword\)](#)
- [VisualElementStyleExtension.Border_Color_Top\(VisualElement, Color\)](#)
- [VisualElementStyleExtension.Border_Color_Top\(VisualElement, StyleKeyword\)](#)
- [VisualElementStyleExtension.Border_Color_Right\(VisualElement, Color\)](#)
- [VisualElementStyleExtension.Border_Color_Right\(VisualElement, StyleKeyword\)](#)
- [VisualElementStyleExtension.Border_Color_Bottom\(VisualElement, Color\)](#)
- [VisualElementStyleExtension.Border_Color_Bottom\(VisualElement, StyleKeyword\)](#)
- [VisualElementStyleExtension.Border_Color_Left\(VisualElement, Color\)](#)
- [VisualElementStyleExtension.Border_Color_Left\(VisualElement, StyleKeyword\)](#)
- [VisualElementStyleExtension.Border_Color\(VisualElement, Color\[\]\)](#)
- [VisualElementStyleExtension.Border_Color\(VisualElement, StyleKeyword\[\]\)](#)
- [VisualElementStyleExtension.GetRadius_Top\(VisualElement\)](#)
- [VisualElementStyleExtension.GetRadius_Bottom\(VisualElement\)](#)
- [VisualElementStyleExtension.GetRadius_Size\(VisualElement\)](#)
- [VisualElementStyleExtension.Border_Radius_Top_Left\(VisualElement, Single, LengthUnit\)](#)
- [VisualElementStyleExtension.Border_Radius_Top_Left\(VisualElement, StyleKeyword\)](#)
- [VisualElementStyleExtension.Border_Radius_Top_Right\(VisualElement, Single, LengthUnit\)](#)
- [VisualElementStyleExtension.Border_Radius_Top_Right\(VisualElement, StyleKeyword\)](#)
- [VisualElementStyleExtension.Border_Radius_Bottom_Left\(VisualElement, Single, LengthUnit\)](#)

VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius(VisualElement, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetBorder_Width(VisualElement)
VisualElementStyleExtension.GetBorder_Height(VisualElement)
VisualElementStyleExtension.GetBorder_Size(VisualElement)
VisualElementStyleExtension.Border_Width_Top(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Right(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Left(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width(VisualElement, Single[])
VisualElementStyleExtension.Border_Width(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Border(VisualElement, Single[])
VisualElementStyleExtension.Border(VisualElement, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)
VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)

VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)

VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)
VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)
VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)
VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)

VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)
VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)
VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)
VisualElementExtension.SetTransform_Position(VisualElement, Vector2)
VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)
VisualElementExtension.GetElementSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)
VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)

CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)
CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32, Boolean)
CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)
CreateElements.CreateElement_Icon(VisualElement, InventoryItem)
CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)
CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class SlotElement.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Inheritance

System.Object

UnityEngine.UIElements.UxmlFactory<[SlotElement](#), [VisualElement.UxmlTraits](#)>

SlotElement.UxmlFactory

Namespace: [InventorySystem.UI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<SlotElement, VisualElement.UxmlTraits>, IUxmlFactory
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)

Class SlotPopupElement

This class is the popup for the Slot, for the splitting and dropping.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement

[PopupBaseElement](#)
SlotPopupElement

Namespace: [InventorySystem.UI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class SlotPopupElement : PopupBaseElement, IEventHandler, ITransform, ITransitionAnimations, IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

SlotPopupElement()

This is the constructor

Declaration

```
public SlotPopupElement()
```

SlotPopupElement(Vector2, Int32, InventoryElement, Boolean)

This is the constructor

Declaration

```
public SlotPopupElement(Vector2 size, int key, InventoryElement From, bool isSplit)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	size	This is the size of the popup.
System.Int32	key	This the key of the item in the dictionary.
InventoryElement	From	This is the From of the dictionary.

TYPE	NAME	DESCRIPTION
System.Boolean	isSplit	This bool will determine if it will split or drop the item.

Methods

GenerateElements()

This will generate the elements for the popup

Declaration

```
public override void GenerateElements()
```

Overrides

[PopupBaseElement.GenerateElements\(\)](#)

Style(Vector2)

This will style the element.

Declaration

```
public override void Style(Vector2 size)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	size	

Overrides

[PopupBaseElement.Style\(Vector2\)](#)

Submit()

This will be invoke by the submit button and will either drop or split the item.

Declaration

```
public override void Submit()
```

Overrides

[PopupBaseElement.Submit\(\)](#)

Extension Methods

VisualElementStyleExtension.BackgroundColor(VisualElement, Color)
VisualElementStyleExtension.BackgroundImage(VisualElement, Background)
VisualElementStyleExtension.BackgroundImage(VisualElement, Texture2D)
VisualElementStyleExtension.BackgroundImage(VisualElement, Sprite)
VisualElementStyleExtension.BackgroundImage(VisualElement, StyleKeyword)
VisualElementStyleExtension.Color(VisualElement, Color)
VisualElementStyleExtension.Color(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Top(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Right(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Left(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color(VisualElement, Color[])
VisualElementStyleExtension.Border_Color(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetRadius_Top(VisualElement)
VisualElementStyleExtension.GetRadius_Bottom(VisualElement)
VisualElementStyleExtension.GetRadius_Size(VisualElement)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius(VisualElement, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetBorder_Width(VisualElement)
VisualElementStyleExtension.GetBorder_Height(VisualElement)
VisualElementStyleExtension.GetBorder_Size(VisualElement)
VisualElementStyleExtension.Border_Width_Top(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Right(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Left(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width(VisualElement, Single[])
VisualElementStyleExtension.Border_Width(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Border(VisualElement, Single[])
VisualElementStyleExtension.Border(VisualElement, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])

VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)
VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)

VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)
VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)

VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)
VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)
VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)
VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)
VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)
VisualElementExtension.SetTransform_Position(VisualElement, Vector2)
VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)

VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)
VisualElementExtension.GetElementSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)
VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)
CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)
CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32, Boolean)
CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)
CreateElements.CreateElement_Icon(VisualElement, InventoryItem)
CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)
CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class SlotPopupElement.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Inheritance

System.Object

UnityEngine.UIElements.UxmlFactory<[SlotPopupElement](#), [VisualElement.UxmlTraits](#)>

SlotPopupElement.UxmlFactory

Namespace: [InventorySystem.UI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<SlotPopupElement, VisualElement.UxmlTraits>, IUxmlFactory
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)

Class TransferPopupElement

This class is the popup for the Slot, for the selling and trading.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement

[PopupBaseElement](#)
TransferPopupElement

Namespace: [InventorySystem.UI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class TransferPopupElement : PopupBaseElement, IEventHandler, ITransform, ITransitionAnimations, IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

TransferPopupElement()

This is the constructor

Declaration

```
public TransferPopupElement()
```

TransferPopupElement(Vector2, TransferItem, InventoryElement, InventoryElement)

This is the constructor

Declaration

```
public TransferPopupElement(Vector2 size, TransferItem item, InventoryElement To, InventoryElement From)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	size	This is the size of the popup.
TransferItem	item	This is the item that will be split or dropped.
InventoryElement	To	This is the purchaser of the item.

TYPE	NAME	DESCRIPTION
InventoryElement	From	This is the owner of the item.

Methods

GenerateElements()

This will generate the elements for the popup

Declaration

```
public override void GenerateElements()
```

Overrides

[PopupBaseElement.GenerateElements\(\)](#)

Style(Vector2)

This will style the element.

Declaration

```
public override void Style(Vector2 size)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	size	

Overrides

[PopupBaseElement.Style\(Vector2\)](#)

Submit()

This will be invoke by the submit button and will either drop or split the item.

Declaration

```
public override void Submit()
```

Overrides

[PopupBaseElement.Submit\(\)](#)

Extension Methods

VisualElementStyleExtension.BackgroundColor(VisualElement, Color)
VisualElementStyleExtension.BackgroundImage(VisualElement, Background)
VisualElementStyleExtension.BackgroundImage(VisualElement, Texture2D)
VisualElementStyleExtension.BackgroundImage(VisualElement, Sprite)
VisualElementStyleExtension.BackgroundImage(VisualElement, StyleKeyword)
VisualElementStyleExtension.Color(VisualElement, Color)
VisualElementStyleExtension.Color(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Top(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Right(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Left(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color(VisualElement, Color[])
VisualElementStyleExtension.Border_Color(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetRadius_Top(VisualElement)
VisualElementStyleExtension.GetRadius_Bottom(VisualElement)
VisualElementStyleExtension.GetRadius_Size(VisualElement)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius(VisualElement, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetBorder_Width(VisualElement)
VisualElementStyleExtension.GetBorder_Height(VisualElement)
VisualElementStyleExtension.GetBorder_Size(VisualElement)
VisualElementStyleExtension.Border_Width_Top(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Right(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Left(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width(VisualElement, Single[])
VisualElementStyleExtension.Border_Width(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Border(VisualElement, Single[])
VisualElementStyleExtension.Border(VisualElement, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])

VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)
VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)

VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)
VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)

VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)
VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)
VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)
VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)
VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)
VisualElementExtension.SetTransform_Position(VisualElement, Vector2)
VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)

VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)
VisualElementExtension.GetElementSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)
VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)
CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)
CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32, Boolean)
CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)
CreateElements.CreateElement_Icon(VisualElement, InventoryItem)
CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)
CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class TransferPopupElement.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Inheritance

System.Object

UnityEngine.UIElements.UxmlFactory<[TransferPopupElement](#), [VisualElement.UxmlTraits](#)>

TransferPopupElement.UxmlFactory

Namespace: [InventorySystem.UI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<TransferPopupElement, VisualElement.UxmlTraits>, IUxmlFactory
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)

Namespace ItemSystem

This is The Namespace for the ItemSystem, Assembly-CSharp

[ItemSystem.Data](#)

[ItemSystem.Editor](#)

Namespace ItemSystem.Data

Classes

[ItemData](#)

This class is a ScriptableObject and is used to represent an item that can be used in the world or in the inventory.

Enums

[ItemType](#)

This Enum is used to determine the type of the Item. To use this enum in UI Builder

[WeaponState](#)

This is used to determine the Current State of the Weapon is in. To use this enum in UI Builder

Class ItemData

This class is a ScriptableObject and is used to represent an item that can be used in the world or in the inventory.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.ScriptableObject
ItemData

Namespace: [ItemSystem.Data](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class ItemData : ScriptableObject
```

Fields

cost

This is the cost of the item.

Declaration

```
public int cost
```

Field Value

TYPE	DESCRIPTION
System.Int32	

description

This is the description of the item.

Declaration

```
public string description
```

Field Value

TYPE	DESCRIPTION
System.String	

IID

This is a unique id for the item. This will be used to identify the InventoryItem and Items in the world.

Declaration

```
public string IID
```

Field Value

TYPE	DESCRIPTION
System.String	

image

This is the Image of the item for the inventory icons.

Declaration

```
public Sprite image
```

Field Value

TYPE	DESCRIPTION
UnityEngine.Sprite	

isEquipable

This will determine if the items can be equipped.

Declaration

```
public bool isEquipable
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

itemName

This is the name of the item.

Declaration

```
public string itemName
```

Field Value

TYPE	DESCRIPTION
System.String	

prefabWorldItem

This is a prefab of the item for 3D games.

Declaration

```
public GameObject prefabWorldItem
```

Field Value

TYPE	DESCRIPTION
UnityEngine.GameObject	

stackable

This determines if the items can be stacked.

Declaration

```
public bool stackable
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

stackSize

This is how many of the item and fit into an inventory stack.

Declaration

```
public int stackSize
```

Field Value

TYPE	DESCRIPTION
System.Int32	

type

This the type of item it is.

Declaration

```
public ItemType type
```

Field Value

TYPE	DESCRIPTION
ItemType	

weight

This is the weight of the item.

Declaration

```
public float weight
```

Field Value

TYPE	DESCRIPTION
System.Single	

Methods

CreateWorldObject(Transform)

This will Instantiate the prefabWorldItem prefab.

Declaration

```
public GameObject CreateWorldObject(Transform target = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	target	The parent of this object.

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
UnityEngine.GameObject	The game object of the item.

Extension Methods

- [RichText.UpperCaseText\(Object\)](#)
- [RichText.LowerCaseText\(Object\)](#)
- [RichText.BoldText\(Object\)](#)
- [RichText.ItalicText\(Object\)](#)
- [RichText.Text\(Object\)](#)
- [RichText.UnderLineText\(Object\)](#)
- [RichText.StrikeText\(Object\)](#)
- [RichText.SupText\(Object\)](#)
- [RichText.SubText\(Object\)](#)
- [RichText.PositionText\(Object, Single\)](#)
- [RichText.SizeText\(Object, Int32\)](#)
- [RichText.ColoredText\(Object, String\)](#)
- [RichText.ColoredText\(Object, Color\)](#)
- [RichText.BooleanText\(Object, Boolean\)](#)
- [RichText.BooleanText\(Object, Boolean, Color\)](#)

Enum ItemType

This Enum is used to determine the type of the Item. To use this enum in UI Builder

Namespace: [ItemSystem.Data](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public enum ItemType
```

Remarks

ItemSystem.Data.ItemType, Assembly-CSharp

Fields

NAME	DESCRIPTION
Armour	
Consumable	
Materials	
Misc	
Weapon	

Extension Methods

- [RichText.UpperCaseText\(\)](#)
- [RichText.LowerCaseText\(\)](#)
- [RichText.BoldText\(\)](#)
- [RichText.ItalicText\(\)](#)
- [RichText.Text\(\)](#)
- [RichText.UnderLineText\(\)](#)
- [RichText.StrikeText\(\)](#)
- [RichText.SupText\(\)](#)
- [RichText.SubText\(\)](#)
- [RichText.PositionText\(Single\)](#)
- [RichText.SizeText\(Int32\)](#)
- [RichText.ColoredText\(String\)](#)
- [RichText.ColoredText\(Color\)](#)
- [RichText.BooleanText\(Boolean\)](#)
- [RichText.BooleanText\(Boolean, Color\)](#)

Enum WeaponState

This is used to determine the Current State of the Weapon is in. To use this enum in UI Builder

Namespace: [ItemSystem.Data](#)

Assembly: [Assembly-CSharp.dll](#)

Syntax

```
public enum WeaponState
```

Remarks

ItemSystem.Data.WeaponState, Assembly-CSharp

Fields

NAME	DESCRIPTION
Attack	
Defend	
Idle	
Parry	

Extension Methods

- [RichText.UpperCaseText\(\)](#)
- [RichText.LowerCaseText\(\)](#)
- [RichText.BoldText\(\)](#)
- [RichText.ItalicText\(\)](#)
- [RichText.Text\(\)](#)
- [RichText.UnderLineText\(\)](#)
- [RichText.StrikeText\(\)](#)
- [RichText.SupText\(\)](#)
- [RichText.SubText\(\)](#)
- [RichText.PositionText\(Single\)](#)
- [RichText.SizeText\(Int32\)](#)
- [RichText.ColoredText\(String\)](#)
- [RichText.ColoredText\(Color\)](#)
- [RichText.BooleanText\(Boolean\)](#)
- [RichText.BooleanText\(Boolean, Color\)](#)

Namespace ItemSystem.Editor

Classes

[CreateItem](#)

This class is an EditorWindow used to create an ItemData file.

[EditItem](#)

This class is an EditorWindow used to edit any ItemData file.

[ItemIDScrubber](#)

This class is an EditorWindow used to check all ItemData in the Resource folder and make sure the ID's are not conflicted.

Class CreateItem

This class is an EditorWindow used to create an ItemData file.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.ScriptableObject
UnityEditor.EditorWindow
CreateItem

Namespace: [ItemSystem.Editor](#)
Assembly: Assembly-CSharp-Editor.dll

Syntax

```
public class CreateItem : EditorWindow
```

Fields

costField

This is the cost that the item will have.

Declaration

```
public IntegerField costField
```

Field Value

TYPE	DESCRIPTION
UnityEditor.UIElements.IntegerField	

createButton

This is the element that will be used to Create the ItemData file and pass in the data from the input fields.

Declaration

```
public Button createButton
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

descriptionField

This is the description of the item will have.

Declaration

```
public TextField descriptionField
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.TextField	

imageField

This is the image that the item will use in the inventory.

Declaration

```
public ObjectField imageField
```

Field Value

TYPE	DESCRIPTION
UnityEditor.UIElements.ObjectField	

itemCanStackToggle

This is to determine if the item will be able to stack.

Declaration

```
public Toggle itemCanStackToggle
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Toggle	

itemNameField

This is the name of the item will have.

Declaration

```
public TextField itemNameField
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.TextField	

itemVisualSprite

This is a display of the image that the item will use.

Declaration

```
public SpriteElement itemVisualSprite
```

Field Value

TYPE	DESCRIPTION
SpriteElement	

newItemData

This is the new ItemData file that is created.

Declaration

```
public ItemData newItemData
```

Field Value

TYPE	DESCRIPTION
ItemData	

OnItemChange

This is the GivelerItem that is subscribed to by the InventoryWindow AddItemData. when the InventoryWindow is open it and an ItemData file is created it will add the new ItemData file ot the AddItemData Method.

Declaration

```
public static Action<ItemData> OnItemChange
```

Field Value

TYPE	DESCRIPTION
System.Action< ItemData >	

stackSizeField

This is the max stack size that the item will have.

Declaration

```
public IntegerField stackSizeField
```

Field Value

TYPE	DESCRIPTION
UnityEditor.UIElements.IntegerField	

type

This is the type that the new item will be.

Declaration

```
public EnumField type
```

Field Value

TYPE	DESCRIPTION
UnityEditor.UIElements.EnumField	

weightField

This is the weight that the item will have.

Declaration

```
public FloatField weightField
```

Field Value

TYPE	DESCRIPTION
UnityEditor.UIElements.FloatField	

Methods

CreateGUI()

This is a unity method, that will be run when the window is created.

Declaration

```
public void CreateGUI()
```

CreateItemFile()

This Will create the item from the inputs you have made.

Declaration

```
public void CreateItemFile()
```

GetUIElements()

This will get the Elements in the UI. And set RegisterValueChangedCallback if any other element's values are changed.

Declaration

```
public void GetUIElements()
```

OpenWindow()

This static will open the CreateItem. Though the UnityEditor Toolbar: Tools/DownUnder Studios/Item System/Tools/Create Item Or by the shortcut key CTRL + SHIFT + ALT + C

Declaration

```
[MenuItem("Tools/DownUnder Studios/Item System/Tools/Create Item %&&C")]  
public static void OpenWindow()
```

ResetField()

This will reset the input field for the item.

Declaration

```
public void ResetField()
```

SetWindowSize(Int32, Int32)

This will set the Editor Window min & max size to a fix size. With both params being the width & height respectively.

Declaration

```
public void SetWindowSize(int width, int height)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	width	The width of the Window.
System.Int32	height	The height of the Window.

Extension Methods

- RichText.UpperCaseText(Object)
- RichText.LowerCaseText(Object)
- RichText.BoldText(Object)
- RichText.ItalicText(Object)
- RichText.Text(Object)
- RichText.UnderLineText(Object)
- RichText.StrikeText(Object)
- RichText.SupText(Object)
- RichText.SubText(Object)
- RichText.PositionText(Object, Single)
- RichText.SizeText(Object, Int32)
- RichText.ColoredText(Object, String)
- RichText.ColoredText(Object, Color)
- RichText.BooleanText(Object, Boolean)
- RichText.BooleanText(Object, Boolean, Color)

Class EditItem

This class is an EditorWindow used to edit any ItemData file.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.ScriptableObject
UnityEditor.EditorWindow
EditItem

Namespace: [ItemSystem.Editor](#)
Assembly: Assembly-CSharp-Editor.dll

Syntax

```
public class EditItem : EditorWindow
```

Fields

costField

This is the element that will be used to get the cost for the item.

Declaration

```
public IntegerField costField
```

Field Value

TYPE	DESCRIPTION
UnityEditor.UIElements.IntegerField	

descriptionField

This is the element that will be used to get the description for the item.

Declaration

```
public TextField descriptionField
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.TextField	

imageField

This is the element that will be used to get the Sprite for the item.

Declaration

```
public ObjectField imageField
```

Field Value

TYPE	DESCRIPTION
UnityEditor.UIElements.ObjectField	

itemCanStackToggle

This is the element that will be used to determine if the item can stack.

Declaration

```
public Toggle itemCanStackToggle
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Toggle	

itemNameField

This is the element that will be used to get the name for the item.

Declaration

```
public TextField itemNameField
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.TextField	

itemVisualSprite

This is the element that will show the item image tat will be used by the inventory.

Declaration

```
public SpriteElement itemVisualSprite
```

Field Value

TYPE	DESCRIPTION
SpriteElement	

stackSizeField

This is the element that will be used to get the max stack size for the item.

Declaration

```
public IntegerField stackSizeField
```

Field Value

TYPE	DESCRIPTION
UnityEditor.UIElements.IntegerField	

targetItemData

This is the ItemData file that will be edited.

Declaration

```
public static ItemData targetItemData
```

Field Value

TYPE	DESCRIPTION
ItemData	

type

This is the element that will be used to get the type of the item.

Declaration

```
public EnumField type
```

Field Value

TYPE	DESCRIPTION
UnityEditor.UIElements.EnumField	

weightField

This is the element that will be used to get the weight of the item.

Declaration

```
public FloatField weightField
```

Field Value

TYPE	DESCRIPTION
UnityEditor.UIElements.FloatField	

Methods

CreateGUI()

This is a unity method, that will be run when the window is created.

Declaration

```
public void CreateGUI()
```

GetUIElements()

This will get the Elements in the UI. And set RegisterValueChangedCallback if any other element's values are changed.

Declaration

```
public void GetUIElements()
```

OnOpenAsset(Int32, Int32)

This will open the EditItem window when an ItemData file is double clicked.

Declaration

```
[OnOpenAsset]  
public static bool OnOpenAsset(int instanceID, int line)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	instanceID	not used.
System.Int32	line	not used.

Returns

TYPE	DESCRIPTION
System.Boolean	not used.

OpenWindow()

This will open the EditItem. The window can be open by double clicking an ItemData file.

Declaration

```
public static void OpenWindow()
```

SetItemData(ItemData)

This will set the target ItemData. and open the window.

Declaration

```
public static bool SetItemData(ItemData item)
```

Parameters

TYPE	NAME	DESCRIPTION
ItemData	item	The ItemData that will be edited.

Returns

TYPE	DESCRIPTION
System.Boolean	Return true if targetItemData is set and window open.

SetWindowSize(Int32, Int32)

This will set the Editor Window min & max size to a fix size. With both params being the width & height respectively.

Declaration

```
public void SetWindowSize(int width, int height)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	width	The width of the Window.
System.Int32	height	The height of the Window.

UpdateItemData()

This will update ItemData file with the new data from the input fields.

Declaration

```
public void UpdateItemData()
```

Extension Methods

- RichText.UpperCaseText(Object)
- RichText.LowerCaseText(Object)
- RichText.BoldText(Object)
- RichText.ItalicText(Object)
- RichText.Text(Object)
- RichText.UnderLineText(Object)
- RichText.StrikeText(Object)
- RichText.SupText(Object)
- RichText.SubText(Object)
- RichText.PositionText(Object, Single)
- RichText.SizeText(Object, Int32)
- RichText.ColoredText(Object, String)
- RichText.ColoredText(Object, Color)
- RichText.BooleanText(Object, Boolean)
- RichText.BooleanText(Object, Boolean, Color)

Class ItemIDScrubber

This class is an EditorWindow used to check all ItemData in the Resource folder and make sure the ID's are not conflicted.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.ScriptableObject
UnityEditor.EditorWindow
ItemIDScrubber

Namespace: [ItemSystem.Editor](#)
Assembly: Assembly-CSharp-Editor.dll

Syntax

```
public class ItemIDScrubber : EditorWindow
```

Fields

clearButton

This is the element that when pressed will clear the Log.

Declaration

```
public Button clearButton
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

currentLogNumber

This is the number of logs made.

Declaration

```
public int currentLogNumber
```

Field Value

TYPE	DESCRIPTION
System.Int32	

items

This is an array of all ItemData in the project.

Declaration

```
public ItemData[] items
```

Field Value

TYPE	DESCRIPTION
ItemData[]	

logDataString

This is the sting that hold all of the logs that will be added into a file upon saving.

Declaration

```
public string logDataString
```

Field Value

TYPE	DESCRIPTION
System.String	

LogEntryParent

This is the parent where all log elements will be parented to.

Declaration

```
public VisualElement LogEntryParent
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.VisualElement	

SaveButton

This is the element that when pressed will save the Log.

Declaration

```
public Button SaveButton
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

startButton

This is the element that when pressed will Start the scrubbing process.

Declaration

```
public Button startButton
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

Methods

Clear()

This will clear the Log.

Declaration

```
public void Clear()
```

CreateGUI()

This is a unity method, that will be run when the window is created.

Declaration

```
public void CreateGUI()
```

CreateLogEntry(String)

This Will create a log entry from a string log.

Declaration

```
public void CreateLogEntry(string log)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	log	this is the log that is passed in.

Save()

This will save the Log to a .txt file. (With time stamp).

Declaration

```
public void Save()
```

SetWindowSize(Int32, Int32)

This will set the Editor Window min & max size to a fix size. With both params being the width & height respectively.

Declaration

```
public void SetWindowSize(int width, int height)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	width	The width of the Window.
System.Int32	height	The height of the Window.

ShowExample()

This will open the CreateItem. Though the UnityEditor Toolbar: Tools/DownUnder Studios/Item System/Tools/Item ID Scrubber Or by the shortcut key CTRL + SHIFT + ALT + S

Declaration

```
[MenuItem("Tools/DownUnder Studios/Item System/Tools/Item ID Scrubber %#&S")]
public static void ShowExample()
```

Start()

This will cross check every ItemData file for conflicting IDs. And return a log for each ID that is changed.

Declaration

```
public void Start()
```

Extension Methods

`RichText.UpperCaseText(Object)`

`RichText.LowerCaseText(Object)`

`RichText.BoldText(Object)`

`RichText.ItalicText(Object)`

`RichText.Text(Object)`

`RichText.UnderLineText(Object)`

`RichText.StrikeText(Object)`

`RichText.SupText(Object)`

`RichText.SubText(Object)`

`RichText.PositionText(Object, Single)`

`RichText.SizeText(Object, Int32)`

`RichText.ColoredText(Object, String)`

`RichText.ColoredText(Object, Color)`

`RichText.BooleanText(Object, Boolean)`

`RichText.BooleanText(Object, Boolean, Color)`

Namespace ItemSystem.Script

Classes

[WorldItem](#)

This abstract class is the base of any item in the world. such as the "PickupItem" or the "EquippedItem".

Class WorldItem

This abstract class is the base of any item in the world. such as the "PickupItem" or the "EquippedItem".

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
WorldItem
[EquippedItem](#)
[PickupItem](#)

Namespace: [ItemSystem.Script](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public abstract class WorldItem : MonoBehaviour
```

Methods

SetItem(ItemData, Int32)

This will set the item and amount.

Declaration

```
public void SetItem(ItemData item, int amount)
```

Parameters

TYPE	NAME	DESCRIPTION
ItemData	item	This is the item to be added.
System.Int32	amount	This is the amount to be added.

Extension Methods

[RichText.UpperCaseText\(Object\)](#)
[RichText.LowerCaseText\(Object\)](#)
[RichText.BoldText\(Object\)](#)
[RichText.ItalicText\(Object\)](#)
[RichText.Text\(Object\)](#)
[RichText.UnderLineText\(Object\)](#)
[RichText.StrikeText\(Object\)](#)
[RichText.SupText\(Object\)](#)

RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)

Namespace UtilitySystem

This is The Namespace for the UtilitySystem, Assembly-CSharp

UtilitySystem.Data

UtilitySystem.Extension

UtilitySystem.Script

UtilitySystem.Tool

UtilitySystem.UI

Namespace UtilitySystem.Data

Classes

[ContextMenuButtonItemData](#)

This class is the ContextMenuItemButtonData. It will be used to create a ContextMenuItemButtonElement. And is a child class of ContextMenuItemData.

[ContextMenuItemData](#)

This class is the ContextMenuItemData. it will be used to create a ContextMenuItemElement.

[ContextMenuItemParentData](#)

This class is the ContextMenuItemParentData. It will be used to create a ContextMenuItemParentElement. And is a child class of ContextMenuItemData.

Enums

[UIAnchorEnum](#)

This is used to determine how an Inventory will be sorted. To use this enum in UI Builder

Class ContextMenuItemButtonData

This class is the ContextMenuItemButtonData. It will be used to create a ContextMenuItemButtonElement. And is a child class of ContextMenuItemData.

Inheritance

System.Object
ContextMenuItemData
ContextMenuItemButtonData

Namespace: [UtilitySystem.Data](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class ContextMenuItemButtonData : ContextMenuItemData
```

Constructors

ContextMenuItemButtonData(String, Action, Boolean)

This is a constructor and will create the Context Menu Item Data. (Button)

Declaration

```
public ContextMenuItemButtonData(string textValue, Action action, bool isSectionEnd = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	textValue	This will be added to the text variable.
System.Action	action	This will be added to the onClick variable.
System.Boolean	isSectionEnd	This will be added to the isSectionEnd variable.

ContextMenuItemButtonData(String, Boolean, Action[])

This is a constructor and will create the Context Menu Item Data. (Button)

Declaration

```
public ContextMenuItemButtonData(string textValue, bool isSectionEnd = false, params Action[] actions)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	textValue	This will be added to the text variable.
System.Boolean	isSectionEnd	This will be added to the isSectionEnd variable.
System.Action[]	actions	This will be added to the onClick variable.

ContextMenuButtonItemButtonData(String, List<Action>, Boolean)

This is a constructor and will create the Context Menu Item Data. (Button)

Declaration

```
public ContextMenuItemButtonData(string textValue, List<Action> actions, bool isSectionEnd = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	textValue	This will be added to the text variable.
System.Collections.Generic.List<System.Action>	actions	This will be added to the onClick variable.
System.Boolean	isSectionEnd	This will be added to the isSectionEnd variable.

Fields

onClick

This will be invoked when the item's is clicked.

Declaration

```
public Action onClick
```

Field Value

TYPE	DESCRIPTION
System.Action	

Extension Methods

`RichText.UpperCaseText(Object)`

`RichText.LowerCaseText(Object)`

`RichText.BoldText(Object)`

`RichText.ItalicText(Object)`

`RichText.Text(Object)`

`RichText.UnderLineText(Object)`

`RichText.StrikeText(Object)`

`RichText.SupText(Object)`

`RichText.SubText(Object)`

`RichText.PositionText(Object, Single)`

`RichText.SizeText(Object, Int32)`

`RichText.ColoredText(Object, String)`

`RichText.ColoredText(Object, Color)`

`RichText.BooleanText(Object, Boolean)`

`RichText.BooleanText(Object, Boolean, Color)`

Class ContextMenuItemData

This class is the ContextMenuItemData. it will be used to create a ContextMenuItemElement.

Inheritance

System.Object
ContextMenuItemData
[ContextMenuItemButtonData](#)
[ContextMenuItemParentData](#)

Namespace: [UtilitySystem.Data](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class ContextMenuItemData
```

Constructors

ContextMenuItemData(String, Boolean)

This is a constructor and will create the Context Menu Item Data.

Declaration

```
public ContextMenuItemData(string textValue, bool isSectionEnd = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	textValue	This will be added to the text variable.
System.Boolean	isSectionEnd	This will be added to the isSectionEnd variable.

Fields

isSectionEnd

If true will add a line breaker after this menu ite.

Declaration

```
public bool isSectionEnd
```

Field Value

TYPE	DESCRIPTION
System.Boolean	

text

This is the text that will be displayed on the item's label.

Declaration

```
public string text
```

Field Value

TYPE	DESCRIPTION
System.String	

Extension Methods

- [RichText.UpperCaseText\(Object\)](#)
- [RichText.LowerCaseText\(Object\)](#)
- [RichText.BoldText\(Object\)](#)
- [RichText.ItalicText\(Object\)](#)
- [RichText.Text\(Object\)](#)
- [RichText.UnderLineText\(Object\)](#)
- [RichText.StrikeText\(Object\)](#)
- [RichText.SupText\(Object\)](#)
- [RichText.SubText\(Object\)](#)
- [RichText.PositionText\(Object, Single\)](#)
- [RichText.SizeText\(Object, Int32\)](#)
- [RichText.ColoredText\(Object, String\)](#)
- [RichText.ColoredText\(Object, Color\)](#)
- [RichText.BooleanText\(Object, Boolean\)](#)
- [RichText.BooleanText\(Object, Boolean, Color\)](#)

Class ContextMenuItemParentData

This class is the ContextMenuItemParentData. It will be used to create a ContextMenuItemParentElement. And is a child class of ContextMenuItemData.

Inheritance

System.Object
[ContextMenuItemData](#)
ContextMenuItemParentData

Namespace: [UtilitySystem.Data](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]
public class ContextMenuItemParentData : ContextMenuItemData
```

Constructors

ContextMenuItemParentData(String, ContextMenuItemData, Boolean)

This is a constructor and will create the Context Menu Item Data. (Menu)

Declaration

```
public ContextMenuItemParentData(string textValue, ContextMenuItemData item, bool isSectionEnd = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	textValue	This will be added to the text variable.
ContextMenuItemData	item	This item will be added to the items variable.
System.Boolean	isSectionEnd	This will be added to the isSectionEnd variable.

ContextMenuItemParentData(String, Boolean, ContextMenuItemData[])

This is a constructor and will create the Context Menu Item Data. (Menu)

Declaration

```
public ContextMenuItemParentData(string textValue, bool isSectionEnd = false, params ContextMenuItemData[] items)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	textValue	This will be added to the text variable.
System.Boolean	isSectionEnd	This will be added to the isSectionEnd variable.
ContextMenuItemData []	items	These params items will be added to the items variable.

ContextMenuItemParentData(String, List<ContextMenuItemData>, Boolean)

This is a constructor and will create the Context Menu Item Data. (Menu)

Declaration

```
public ContextMenuItemParentData(string textValue, List<ContextMenuItemData> items, bool isSectionEnd = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	textValue	This will be added to the text variable.
System.Collections.Generic.List< ContextMenuItemData >	items	These items will be added to the items variable.
System.Boolean	isSectionEnd	This will be added to the isSectionEnd variable.

Fields

items

This is the list of Context Menu Item Data that will be used to create a sub menu inside of the menu.

Declaration

```
public List<ContextMenuItemData> items
```

Field Value

TYPE	DESCRIPTION
System.Collections.Generic.List< ContextMenuItemData >	

Extension Methods

- [RichText.UpperCaseText\(Object\)](#)
- [RichText.LowerCaseText\(Object\)](#)
- [RichText.BoldText\(Object\)](#)
- [RichText.ItalicText\(Object\)](#)
- [RichText.Text\(Object\)](#)
- [RichText.UnderLineText\(Object\)](#)
- [RichText.StrikeText\(Object\)](#)
- [RichText.SupText\(Object\)](#)
- [RichText.SubText\(Object\)](#)
- [RichText.PositionText\(Object, Single\)](#)
- [RichText.SizeText\(Object, Int32\)](#)
- [RichText.ColoredText\(Object, String\)](#)
- [RichText.ColoredText\(Object, Color\)](#)
- [RichText.BooleanText\(Object, Boolean\)](#)
- [RichText.BooleanText\(Object, Boolean, Color\)](#)

Enum UIAnchorEnum

This is used to determine how an Inventory will be sorted. To use this enum in UI Builder

Namespace: [UtilitySystem.Data](#)

Assembly: [Assembly-CSharp.dll](#)

Syntax

```
public enum UIAnchorEnum
```

Remarks

UtilitySystem.Data.SortByEnum, Assembly-CSharp

Fields

NAME	DESCRIPTION
Bottom_Center	
Bottom_Left	
Bottom_Right	
Center_Center	
Center_Left	
Center_Right	
None	
Top_Center	
Top_Left	
Top_Right	

Extension Methods

- [RichText.UpperCaseText\(\)](#)
- [RichText.LowerCaseText\(\)](#)
- [RichText.BoldText\(\)](#)
- [RichText.ItalicText\(\)](#)
- [RichText.Text\(\)](#)
- [RichText.UnderLineText\(\)](#)
- [RichText.StrikeText\(\)](#)
- [RichText.SupText\(\)](#)
- [RichText.SubText\(\)](#)
- [RichText.PositionText\(Single\)](#)

RichText.SizeType(Int32)

RichText.ColoredText(String)

RichText.ColoredText(Color)

RichText.BooleanText(Boolean)

RichText.BooleanText(Boolean, Color)

Namespace UtilitySystem.Extension

Classes

[FloatExtension](#)

This static class extends the float Variable.

[IntExtension](#)

This static class extends the integer Variable.

[TransformExtension](#)

This static class extends the Transform type.

[VectorExtension](#)

This static class extends the Vector2,3,4 types.

[VisualElementExtension](#)

This static class is used to create short-hand extensions that will cut down on style boilerplate for the VisualElement(s)

[VisualElementStyleExtension](#)

This static class is the extensions of the VisualElement. That deals with the boilerplate and simplify the styles.

Class FloatExtension

This static class extends the float Variable.

Inheritance

System.Object
FloatExtension

Namespace: [UtilitySystem.Extension](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public static class FloatExtension
```

Methods

FlipValue(Single)

This will change an negative number to positive and positive to negative.

Declaration

```
public static float FlipValue(this float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	the float that will be flipped.

Returns

TYPE	DESCRIPTION
System.Single	return the flipped float.

HighestValue(Single, Single)

This will take the highest of 2 floats.

Declaration

```
public static float HighestValue(this float value, float compare)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	The float that will check.
System.Single	compare	The float that will be compare.

Returns

TYPE	DESCRIPTION
System.Single	Return the highest float.

LowestValue(Single, Single)

This will take the lowest of 2 floats.

Declaration

```
public static float LowestValue(this float value, float compare)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	The float that will check.
System.Single	compare	The float that will be compare.

Returns

TYPE	DESCRIPTION
System.Single	Return the lowest float.

RoundDown(Single)

This will round down the float to an integer.

Declaration

```
public static int RoundDown(this float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	the float that will be round down.

Returns

TYPE	DESCRIPTION
System.Int32	Return an float of the float round down.

RoundUp(Single)

This will round up the float to an integer.

Declaration

```
public static int RoundUp(this float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	the float that will be round up.

Returns

TYPE	DESCRIPTION
System.Int32	Return an float of the float round up.

WeightGramToKilogram(Single, Int32)

This will divide the value by 1000

Declaration

```
public static float WeightGramToKilogram(this float value, int decimalPoint = 2)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	The Grams that will converted to Kilograms.
System.Int32	decimalPoint	How many Decimal places.

Returns

TYPE	DESCRIPTION
System.Single	Return the Kilograms.

Remarks

1 Grams is the same as 0.001 Kilogram.

WeightGramToOunce(Single, Int32)

This will divide the value by 28.35

Declaration

```
public static float WeightGramToOunce(this float value, int decimalPoint = 2)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	The Pound that will converted to Ounce.
System.Int32	decimalPoint	How many Decimal places.

Returns

TYPE	DESCRIPTION
System.Single	Return the Ounce.

Remarks

1 Grams is the same as 0.035274 Ounce.

WeightKilogramToGram(Single, Int32)

This will multiple the value by 1000

Declaration

```
public static float WeightKilogramToGram(this float value, int decimalPoint = 2)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	The kilograms that will converted to Grams.
System.Int32	decimalPoint	How many Decimal places.

Returns

TYPE	DESCRIPTION
System.Single	Return the Grams.

Remarks

1 Kilogram is the same as 1000 Grams.

WeightKiloGramToPound(Single, Int32)

This will divide the value by 2.205

Declaration

```
public static float WeightKiloGramToPound(this float value, int decimalPoint = 2)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	The Kilogram that will converted to Pound.
System.Int32	decimalPoint	How many Decimal places.

Returns

TYPE	DESCRIPTION
System.Single	Return the Pound.

Remarks

1 Kilogram is the same as 2.20462 Pound.

WeightOunceToGram(Single, Int32)

This will multiple the value by 28.35

Declaration

```
public static float WeightOunceToGram(this float value, int decimalPoint = 2)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	The Ounce that will converted to Gram.
System.Int32	decimalPoint	How many Decimal places.

Returns

TYPE	DESCRIPTION
System.Single	Return the Gram.

Remarks

1 Ounce is the same as 28.3495 Gram.

WeightOunceToPound(Single, Int32)

This will divide the value by 16

Declaration

```
public static float WeightOunceToPound(this float value, int decimalPoint = 2)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	The Ounce that will converted to Pound.
System.Int32	decimalPoint	How many Decimal places.

Returns

TYPE	DESCRIPTION
System.Single	Return the Pound.

Remarks

1 Ounce is the same as 0.0625 Pound.

WeightPoundToKiloGram(Single, Int32)

This will multiple the value by 2.205

Declaration

```
public static float WeightPoundToKiloGram(this float value, int decimalPoint = 2)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	The Pound that will converted to Kilogram.
System.Int32	decimalPoint	How many Decimal places.

Returns

TYPE	DESCRIPTION
System.Single	Return the Kilogram.

Remarks

1 Pound is the same as 0.453592 Kilogram.

WeightPoundToOunce(Single, Int32)

This will multiple the value by 16

Declaration

```
public static float WeightPoundToOunce(this float value, int decimalPoint = 2)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	The Pound that will converted to Ounce.
System.Int32	decimalPoint	How many Decimal places.

Returns

TYPE	DESCRIPTION
System.Single	Return the Ounce.

Remarks

1 Pound is the same as 16 Ounce.

Class IntExtension

This static class extends the integer Variable.

Inheritance

System.Object
IntExtension

Namespace: [UtilitySystem.Extension](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public static class IntExtension
```

Methods

CurrencyValue(Int32)

This will split the value into gold, silver & bronze.

Declaration

```
public static (int, int, int) CurrencyValue(this int value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	value	The currency amount.

Returns

TYPE	DESCRIPTION
System.ValueTuple<System.Int32, System.Int32, System.Int32>	Return the gold,silver,bronze value.

Remarks

1 gold is 100 silver, 1 silver is 100 copper.

FlipValue(Int32)

This will change an negative number to positive and positive to negative.

Declaration

```
public static int FlipValue(this int value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	value	the integer that will be flipped.

Returns

TYPE	DESCRIPTION
System.Int32	return the flipped integer.

HighestValue(Int32, Int32)

This will take the highest of 2 integers.

Declaration

```
public static int HighestValue(this int value, int compare)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	value	The integer that will check.
System.Int32	compare	The integer that will be compare.

Returns

TYPE	DESCRIPTION
System.Int32	Return the highest integer.

LowestValue(Int32, Int32)

This will take the lowest of 2 integers.

Declaration

```
public static int LowestValue(this int value, int compare)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	value	The integer that will check.
System.Int32	compare	The integer that will be compare.

Returns

TYPE	DESCRIPTION
System.Int32	Return the lowest integer.

Class TransformExtension

This static class extends the Transform type.

Inheritance

System.Object
TransformExtension

Namespace: [UtilitySystem.Extension](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public static class TransformExtension
```

Methods

GetChildren(Transform, Boolean)

This will get a list of the children from a gameobject.

Declaration

```
public static List<GameObject> GetChildren(this Transform transform, bool includeInactive = false)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	This is the transform that will get it's children.
System.Boolean	includeInactive	If the param is true it will also includeInactive gameobject. else ignore.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.List<UnityEngine.GameObject>	return a list of GameObjects.

ResetTransformation(Transform)

This will reset the transform.

Declaration

```
public static void ResetTransformation(this Transform transform)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Transform	transform	This is the transform that will get reset.

Class VectorExtension

This static class extends the Vector2,3,4 types.

Inheritance

System.Object
VectorExtension

Namespace: [UtilitySystem.Extension](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public static class VectorExtension
```

Methods

ClampVector2(Vector2, Vector2, Vector2)

This will clamp a Vector2 between 2 other Vector2s

Declaration

```
public static Vector2 ClampVector2(this Vector2 value, Vector2 min, Vector2 max)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	value	This is the Vector2 that will be clamped.
UnityEngine.Vector2	min	This is the Vector2 that is the min value.
UnityEngine.Vector2	max	This is the Vector2 that is the max value.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	The return is the Vector2 clamped.

ClampVector3(Vector3, Vector3, Vector3)

This will clamp a Vector3 between 2 other Vector3s

Declaration

```
public static Vector3 ClampVector3(this Vector3 value, Vector3 min, Vector3 max)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector3	value	This is the Vector3 that will be clamped.
UnityEngine.Vector3	min	This is the Vector3 that is the min value.
UnityEngine.Vector3	max	This is the Vector3 that is the max value.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector3	The return is the Vector3 clamped.

ClampVector4(Vector4, Vector4, Vector4)

This will clamp a Vector4 between 2 other Vector4s

Declaration

```
public static Vector4 ClampVector4(this Vector4 value, Vector4 min, Vector4 max)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector4	value	This is the Vector4 that will be clamped.
UnityEngine.Vector4	min	This is the Vector4 that is the min value.
UnityEngine.Vector4	max	This is the Vector4 that is the max value.

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
UnityEngine.Vector4	The return is the Vector4 clamped.

HighestVector2(Vector2, Vector2)

This will get the highest Vector2 between 2 Vector2s

Declaration

```
public static Vector2 HighestVector2(this Vector2 value, Vector2 compare)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	value	This is the Vector2 that will be clamped.
UnityEngine.Vector2	compare	This is the Vector2 that will be compared to.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	The return is the Vector2 that is the highest.

HighestVector3(Vector3, Vector3)

This will get the highest Vector3 between 2 Vector3s

Declaration

```
public static Vector2 HighestVector3(this Vector3 value, Vector3 compare)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector3	value	This is the Vector3 that will be clamped.
UnityEngine.Vector3	compare	This is the Vector3 that will be compared to.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	The return is the Vector3 that is the highest.

HighestVector4(Vector4, Vector4)

This will get the highest Vector4 between 2 Vector4s

Declaration

```
public static Vector4 HighestVector4(this Vector4 value, Vector4 compare)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector4	value	This is the Vector4 that will be clamped.
UnityEngine.Vector4	compare	This is the Vector4 that will be compared to.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector4	The return is the Vector4 that is the highest.

LowestVector2(Vector2, Vector2)

This will get the lowest Vector2 between 2 Vector2s

Declaration

```
public static Vector2 LowestVector2(this Vector2 value, Vector2 compare)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	value	This is the Vector2 that will be clamped.

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	compare	This is the Vector2 that will be compared to.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	The return is the Vector2 that is the lowest.

LowestVector3(Vector3, Vector3)

This will get the lowest Vector3 between 2 Vector3s

Declaration

```
public static Vector2 LowestVector3(this Vector3 value, Vector3 compare)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector3	value	This is the Vector3 that will be clamped.
UnityEngine.Vector3	compare	This is the Vector3 that will be compared to.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	The return is the Vector3 that is the lowest.

LowestVector4(Vector4, Vector4)

This will get the lowest Vector4 between 2 Vector4s

Declaration

```
public static Vector2 LowestVector4(this Vector4 value, Vector4 compare)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector4	value	This is the Vector4 that will be clamped.
UnityEngine.Vector4	compare	This is the Vector4 that will be compared to.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	The return is the Vector4 that is the lowest.

Class VisualElementExtension

This static class is used to create short-hand extensions that will cut down on style boilerplate for the VisualElement(s)

Inheritance

System.Object

VisualElementExtension

Namespace: [UtilitySystem.Extension](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public static class VisualElementExtension
```

Methods

Child(VisualElement, VisualElement, Boolean)

This will add, remove an element in this parent or clear the children from the parent.

Declaration

```
public static void Child(this VisualElement parent, VisualElement content = null, bool isAdd = true)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	The parent that will add, remove a child element or clear all children.
UnityEngine.UIElements.VisualElement	content	This is an optional param if not null will allow for adding or remove of the content element. if null them clear all of the children from the element.
System.Boolean	isAdd	this is an optional param the will determine of the stylesheet will be added or remove from the element.

Focus(VisualElement, Boolean)

This will change the focusable from value.

Declaration

```
public static void Focus(this VisualElement element, bool value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's focusable changed.
System.Boolean	value	The boolean that will change value.

Focus_OFF(VisualElement)

This will not allow this element to be interactable

Declaration

```
public static void Focus_OFF(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's focusable set to false.

Focus_ON(VisualElement)

This will allow this element to be interactable.

Declaration

```
public static void Focus_ON(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's focusable set to true.

Focus_Toggle(VisualElement)

This will toggle the focusable from true to false and vis vera.

Declaration

```
public static void Focus_Toggle(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's focusable changed.

GetElementSize(VisualElement)

This will get the size of the element from the largest of the size.

Declaration

```
public static Vector2 GetElementSize(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will check it's size.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	The return will be of the largest width and height of the element.

Remarks

this is used by ScreenBound method to keep the element within the screen.

GetLayout(VisualElement)

This will set the element layout.

Declaration

```
public static Rect GetLayout(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will set the width and height of the element.

Returns

TYPE	DESCRIPTION
UnityEngine.Rect	

GetLayoutHeight(VisualElement)

This will set the size of the element layout.

Declaration

```
public static float GetLayoutHeight(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will set the width and height of the element.

Returns

TYPE	DESCRIPTION
System.Single	

GetLayoutPosition(VisualElement)

This will set the position of the element layout.

Declaration

```
public static Vector2 GetLayoutPosition(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will set the width and height of the element.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	

GetLayoutPositionX(VisualElement)

This will set the position of the element layout.

Declaration

```
public static float GetLayoutPositionX(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will set the width and height of the element.

Returns

TYPE	DESCRIPTION
System.Single	

GetLayoutPositionY(VisualElement)

This will set the position of the element layout.

Declaration

```
public static float GetLayoutPositionY(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will set the width and height of the element.

Returns

TYPE	DESCRIPTION
System.Single	

GetLayoutSize(VisualElement)

This will set the size of the element layout.

Declaration

```
public static Vector2 GetLayoutSize(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will set the width and height of the element.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	

GetLayoutWidth(VisualElement)

This will set the size of the element layout.

Declaration

```
public static float GetLayoutWidth(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will set the width and height of the element.

Returns

TYPE	DESCRIPTION
System.Single	

GetLocalBound(VisualElement)

This will get the element LocalBound.

Declaration

```
public static Rect GetLocalBound(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will set the width and height of the element.

Returns

TYPE	DESCRIPTION
UnityEngine.Rect	

GetLocalBoundPosition(VisualElement)

This will set the position of the element LocalBound.

Declaration

```
public static Vector2 GetLocalBoundPosition(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will set the width and height of the element.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	

GetLocalBoundSize(VisualElement)

This will get the width and high of the element LocalBound.

Declaration

```
public static Vector2 GetLocalBoundSize(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will set the width and height of the element.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	

GetTransform_Position(VisualElement)

This will get the transform scale.

Declaration

```
public static Vector2 GetTransform_Position(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will get it's transform position.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	The return is the transform position.

GetTransform_Rotate(VisualElement)

Declaration

```
public static Quaternion GetTransform_Rotate(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	

Returns

TYPE	DESCRIPTION
UnityEngine.Quaternion	

GetTransform_Scale(VisualElement)

This will get the transform scale.

Declaration

```
public static Vector2 GetTransform_Scale(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will get it's transform scale.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	The return is the transform scale.

GetVisible(VisualElement)

This will get this element's visible.

Declaration

```
public static bool GetVisible(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's visible changed.

Returns

TYPE	DESCRIPTION
System.Boolean	

GetWorldBound(VisualElement)

This will get the element WorldBound.

Declaration

```
public static Rect GetWorldBound(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will set the width and height of the element.

Returns

TYPE	DESCRIPTION
UnityEngine.Rect	

GetWorldBoundPosition(VisualElement)

This will set the position of the element WorldBound.

Declaration

```
public static Vector2 GetWorldBoundPosition(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will set the width and height of the element.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	

GetWorldBoundSize(VisualElement)

This will get the width and high of the element WorldBound.

Declaration

```
public static Vector2 GetWorldBoundSize(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will set the width and height of the element.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	

MouseGrab(VisualElement, Vector2)

This will prepare the element for it being grabbed.

Declaration

```
public static Vector2 MouseGrab(this VisualElement element, Vector2 position)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will be grabbed by mouse.
UnityEngine.Vector2	position	This is the position of the element.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	The return is the offset of the mouse and element position.

MouseRelease(VisualElement)

This will release the element.

Declaration

```
public static Vector2 MouseRelease(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will be used to get the screen bounds.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	The return is the new position of the element.

Name(VisualElement, String)

This will set the elements' name.

Declaration

```
public static void Name(this VisualElement element, string name)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will have a name applied to it.
System.String	name	This is the name that will be applied to the element.

Remarks

the name will be structured as "*Type_name*" with *name* replacing spacing with '-'.

Pick(VisualElement, PickingMode)

This will change the pick mode from value.

Declaration

```
public static void Pick(this VisualElement element, PickingMode value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's pick mode changed.
UnityEngine.UIElements.PickingMode	value	The PickingMode that will change value.

Pick_OFF(VisualElement)

This will not allow this element to be interactable

Declaration

```
public static void Pick_OFF(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's pick mode set to ignore.

Pick_ON(VisualElement)

This will allow this element to be interactable.

Declaration

```
public static void Pick_ON(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's pick mode set to position.

Pick_Toggle(VisualElement)

This will change the pick mode from position to ignore and vis vera.

Declaration

```
public static void Pick_Toggle(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's pick mode changed.

ResetRotation(VisualElement)

This will reset the rotation of the UI element.

Declaration

```
public static void ResetRotation(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's rotation reset.

Rich_Text(TextElement, Boolean)

This will set the TextElement's enableRichText.

Declaration

```
public static void Rich_Text(this TextElement element, bool value = true)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.TextElement	element	This is the element that will be changed.
System.Boolean	value	This value will be applied to the element.

Rotate_Clockwise(VisualElement, Vector3)

This will set the rotation of the UI element on z-axis Counter-Clockwise.

Declaration

```
public static void Rotate_Clockwise(this VisualElement element, Vector3 value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's rotation set.
UnityEngine.Vector3	value	

Rotate_Clockwise_X(VisualElement, Single)

This will increase the rotation of the UI element on z-axis Clockwise.

Declaration

```
public static void Rotate_Clockwise_X(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's rotation changed.
System.Single	value	

Rotate_Clockwise_Z(VisualElement, Single)

This will increase the rotation of the UI element on z-axis Clockwise.

Declaration

```
public static void Rotate_Clockwise_Z(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's rotation changed.
System.Single	value	

Rotate_Counter_Clockwise(VisualElement, Vector3)

This will set the rotation of the UI element on z-axis Counter-Clockwise.

Declaration

```
public static void Rotate_Counter_Clockwise(this VisualElement element, Vector3 value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's rotation set.
UnityEngine.Vector3	value	

Rotate_Counter_Clockwise_X(VisualElement, Single)

This will increase the rotation of the UI element on z-axis Counter-Clockwise.

Declaration

```
public static void Rotate_Counter_Clockwise_X(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's rotation changed.
System.Single	value	

Rotate_Counter_Clockwise_Y(VisualElement, Single)

This will increase the rotation of the UI element on z-axis Counter-Clockwise.

Declaration

```
public static void Rotate_Counter_Clockwise_Y(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's rotation changed.
System.Single	value	

Rotate_Counter_Clockwise_Z(VisualElement, Single)

This will increase the rotation of the UI element on z-axis Counter-Clockwise.

Declaration

```
public static void Rotate_Counter_Clockwise_Z(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's rotation changed.
System.Single	value	

RotateY_Clockwise_Y(VisualElement, Single)

This will increase the rotation of the UI element on z-axis Clockwise.

Declaration

```
public static void RotateY_Clockwise_Y(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's rotation changed.
System.Single	value	

ScreenBound(VisualElement)

This will return the screen bound for the element, Should be used to clamp the element to the screen so it doesn't go off screen.

Declaration

```
public static Vector2 ScreenBound(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be used to determine that screen size - the element.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	The return of the screen bound.

SetTransform_Position(VisualElement, Vector2)

This will set the transform position.

Declaration

```
public static void SetTransform_Position(this VisualElement element, Vector2 value)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's transform position changed.
UnityEngine.Vector2	value	this is the value that will be apply to the transform position.

SetTransform_Rotate(VisualElement, Vector3)

This will set the rotation of the UI element on z-axis Counter-Clockwise.

Declaration

```
public static void SetTransform_Rotate(this VisualElement element, Vector3 value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's rotation set.
UnityEngine.Vector3	value	

SetTransform_Scale(VisualElement, Vector3)

This will set the transform scale.

Declaration

```
public static void SetTransform_Scale(this VisualElement element, Vector3 value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's transform scale changed.
UnityEngine.Vector3	value	this is the value that will be apply to the transform scale.

SetVisible(VisualElement, Boolean)

This will set the visible from value.

Declaration

```
public static void SetVisible(this VisualElement element, bool value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's visible changed.
System.Boolean	value	The boolean that will change value.

StyleSheet(VisualElement, StyleSheet, Boolean)

This will add, remove a stylesheet in this parent or clear the stylesheets in the element.

Declaration

```
public static void StyleSheet(this VisualElement element, StyleSheet style = null, bool isAdd = true)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will a style sheet added, removed or cleared.
UnityEngine.UIElements.StyleSheet	style	This is an optional param if not null will allow for adding or remove of this stylesheet. if null them clear all of the stylesheets from the element.
System.Boolean	isAdd	this is an optional param the will determine of the stylesheet will be added or remove from the element. this is an optional param the will determine of the stylesheet will be added or remove from the element.

ToolTip(VisualElement, String)

This is the tooltip of the element.

Declaration

```
public static void ToolTip(this VisualElement element, string value)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's tooltip changed.
System.String	value	This is the string that will be displayed on the tooltip.

UpdatePosition(VisualElement, Vector2)

This will move the element to the mouse position - the offset.

Declaration

```
public static void UpdatePosition(this VisualElement element, Vector2 offset)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will be used to get the screen bounds.
UnityEngine.Vector2	offset	The offset is the distance of the initial mouse grab. to the element position.

UsageHints(VisualElement, UsageHints)

This is the usageHints of the element.

Declaration

```
public static void UsageHints(this VisualElement element, UsageHints value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's usageHints changed.
UnityEngine.UIElements.UsageHints	value	This is the UsageHints for the element.

ViewDataKey(VisualElement, String)

This is the viewDataKey of the element.

Declaration

```
public static void ViewDataKey(this VisualElement element, string value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's viewDataKey changed.
System.String	value	This is the data for the viewDataKey.

Visible_Hide(VisualElement)

This will hide this element but keep it's effects on the UI.

Declaration

```
public static void Visible_Hide(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's visible set to false.

Visible_IsInvisible(VisualElement)

This will get the element's visible

Declaration

```
public static bool Visible_IsInvisible(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be checked.

Returns

TYPE	DESCRIPTION
System.Boolean	return true if display is false.

Visible_IsVisible(VisualElement)

This will get the element's visible

Declaration

```
public static bool Visible_IsVisible(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be checked.

Returns

TYPE	DESCRIPTION
System.Boolean	return true if display is true.

Visible_Show(VisualElement)

This will show this element and keep it's effects on the UI.

Declaration

```
public static void Visible_Show(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's visible set to true.

Visible_ToggleVisibility(VisualElement)

This will toggle the visible from true to false and vis vera.

Declaration

```
public static void Visible_ToggleVisibility(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's visible changed.

Class VisualElementStyleExtension

This static class is the extensions of the VisualElement. That deals with the boilerplate and simplify the styles.

Inheritance

System.Object
VisualElementStyleExtension

Namespace: [UtilitySystem.Extension](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public static class VisualElementStyleExtension
```

Methods

Align_Content(VisualElement, Align)

This will set the element's style alignContent

Declaration

```
public static void Align_Content(this VisualElement element, Align value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.Align	value	This value will be applied to the element.

Align_Content(VisualElement, StyleKeyword)

This will set the element's style alignContent

Declaration

```
public static void Align_Content(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Align_Item(VisualElement, Align)

This will set the element's style alignItems

Declaration

```
public static void Align_Item(this VisualElement element, Align value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.Align	value	This value will be applied to the element.

Align_Item(VisualElement, StyleKeyword)

This will set the element's style alignItems

Declaration

```
public static void Align_Item(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Align_Self(VisualElement, Align)

This will set the element's style alignSelf

Declaration

```
public static void Align_Self(this VisualElement element, Align value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.Align	value	This value will be applied to the element.

Align_Self(VisualElement, StyleKeyword)

This will set the element's style alignSelf

Declaration

```
public static void Align_Self(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)

This will set the element's style Bottom & Left.

Declaration

```
public static void Anchored_Position_Bottom_Left(this VisualElement element, UIAnchorEnum value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UIAnchorEnum	value	This value will be applied to the element.

Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)

This will set the element's style Bottom & Right.

Declaration

```
public static void Anchored_Position_Bottom_Right(this VisualElement element, UIAnchorEnum value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UIAnchorEnum	value	This value will be applied to the element.

Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)

This will set the element's style Top & Left.

Declaration

```
public static void Anchored_Position_Top_Left(this VisualElement element, UIAnchorEnum value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UIAnchorEnum	value	This value will be applied to the element.

Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)

This will set the element's style Top & Right.

Declaration

```
public static void Anchored_Position_Top_Right(this VisualElement element, UIAnchorEnum value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

TYPE	NAME	DESCRIPTION
UIAnchorEnum	value	This value will be applied to the element.

BackGround_Color(VisualElement, Color)

This will set the element's style

Declaration

```
public static void BackGround_Color(this VisualElement element, Color value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.Color	value	This value will be applied to the element.

Background_Image(VisualElement, Sprite)

This will set the element's style

Declaration

```
public static void Background_Image(this VisualElement element, Sprite value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.Sprite	value	This value will be applied to the element.

Background_Image(VisualElement, Texture2D)

This will set the element's style

Declaration

```
public static void Background_Image(this VisualElement element, Texture2D value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.Texture2D	value	This value will be applied to the element.

Background_Image(VisualElement, Background)

This will set the element's style

Declaration

```
public static void Background_Image(this VisualElement element, Background value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.Background	value	This value will be applied to the element.

Background_Image(VisualElement, StyleKeyword)

This will set the element's style

Declaration

```
public static void Background_Image(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Border(VisualElement, Single[])

This will set the element's style Border Width, Color & Radius.

Declaration

```
public static void Border(this VisualElement element, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single[]	args	These value will be applied to the element.

Border(VisualElement, Color, Single[])

This will set the element's style Border Width, Color & Radius.

Declaration

```
public static void Border(this VisualElement element, Color color, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.Color	color	This color that will be applied to the element's border.
System.Single[]	args	These value will be applied to the element.

Border(VisualElement, Color[], Single[])

This will set the element's style Border Width, Color & Radius.

Declaration

```
public static void Border(this VisualElement element, Color[] color, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.Color[]	color	This color that will be applied to the element's border.
System.Single[]	args	These value will be applied to the element.

Border(VisualElement, LengthUnit, Color, Single[])

This will set the element's style Border Width, Color & Radius.

Declaration

```
public static void Border(this VisualElement element, LengthUnit unit, Color color, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.
UnityEngine.Color	color	This color that will be applied to the element's border.
System.Single[]	args	These value will be applied to the element.

Border(VisualElement, LengthUnit, Color[], Single[])

This will set the element's style Border Width, Color & Radius.

Declaration

```
public static void Border(this VisualElement element, LengthUnit unit, Color[] color, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.
UnityEngine.Color[]	color	This color that will be applied to the element's border.
System.Single[]	args	These value will be applied to the element.

Border(VisualElement, LengthUnit[], Color, Single[])

This will set the element's style Border Width, Color & Radius.

Declaration

```
public static void Border(this VisualElement element, LengthUnit[] unit, Color color, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.LengthUnit[]	unit	This unit will be applied to the element.
UnityEngine.Color	color	This color that will be applied to the element's border.
System.Single[]	args	These value will be applied to the element.

Border(VisualElement, LengthUnit[], Color[], Single[])

This will set the element's style Border Width, Color & Radius.

Declaration

```
public static void Border(this VisualElement element, LengthUnit[] unit, Color[] color, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.LengthUnit[]	unit	This unit will be applied to the element.
UnityEngine.Color[]	color	This color that will be applied to the element's border.
System.Single[]	args	These value will be applied to the element.

Border(VisualElement, StyleKeyword[])

This will set the element's style Border Width & Radius.

Declaration

```
public static void Border(this VisualElement element, params StyleKeyword[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword[]	args	These value will be applied to the element.

Border_Color(VisualElement, Color[])

This will set the element's style borderTopColor, borderRightColor, borderBottomColor & borderLeftColor.

Declaration

```
public static void Border_Color(this VisualElement element, params Color[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

TYPE	NAME	DESCRIPTION
UnityEngine.Color[]	args	These value will be applied to the element.

Border_Color(VisualElement, StyleKeyword[])

This will set the element's style borderTopColor, borderRightColor, borderBottomColor & borderLeftColor.

Declaration

```
public static void Border_Color(this VisualElement element, params StyleKeyword[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword[]	args	These value will be applied to the element.

Border_Color_Bottom(VisualElement, Color)

This will set the element's style borderBottomColor.

Declaration

```
public static void Border_Color_Bottom(this VisualElement element, Color value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.Color	value	This value will be applied to the element.

Border_Color_Bottom(VisualElement, StyleKeyword)

This will set the element's style borderBottomColor.

Declaration

```
public static void Border_Color_Bottom(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Border_Color_Left(VisualElement, Color)

This will set the element's style borderLeftColor.

Declaration

```
public static void Border_Color_Left(this VisualElement element, Color value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.Color	value	This value will be applied to the element.

Border_Color_Left(VisualElement, StyleKeyword)

This will set the element's style borderLeftColor.

Declaration

```
public static void Border_Color_Left(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Border_Color_Right(VisualElement, Color)

This will set the element's style borderRightColor.

Declaration

```
public static void Border_Color_Right(this VisualElement element, Color value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.Color	value	This value will be applied to the element.

Border_Color_Right(VisualElement, StyleKeyword)

This will set the element's style `borderRightColor`.

Declaration

```
public static void Border_Color_Right(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Border_Color_Top(VisualElement, Color)

This will set the element's style `borderTopColor`.

Declaration

```
public static void Border_Color_Top(this VisualElement element, Color value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

TYPE	NAME	DESCRIPTION
UnityEngine.Color	value	This value will be applied to the element.

Border_Color_Top(VisualElement, StyleKeyword)

This will set the element's style borderTopColor.

Declaration

```
public static void Border_Color_Top(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Border_Radius(VisualElement, Single[])

This will set the element's style borderTopLeftRadius, borderTopRightRadius, borderBottomLeftRadius & borderBottomRightRadius.

Declaration

```
public static void Border_Radius(this VisualElement element, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single[]	args	These value will be applied to the element.

Border_Radius(VisualElement, LengthUnit, Single[])

This will set the element's style borderTopLeftRadius, borderTopRightRadius, borderBottomLeftRadius & borderBottomRightRadius.

Declaration

```
public static void Border_Radius(this VisualElement element, LengthUnit unit, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.
System.Single[]	args	These value will be applied to the element.

Border_Radius(VisualElement, LengthUnit[], Single[])

This will set the element's style `borderTopLeftRadius`, `borderTopRightRadius`, `borderBottomLeftRadius` & `borderBottomRightRadius`.

Declaration

```
public static void Border_Radius(this VisualElement element, LengthUnit[] unit, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.LengthUnit[]	unit	This unit will be applied to the element.
System.Single[]	args	These value will be applied to the element.

Border_Radius(VisualElement, StyleKeyword[])

This will set the element's style `borderTopLeftRadius`, `borderTopRightRadius`, `borderBottomLeftRadius` & `borderBottomRightRadius`.

Declaration

```
public static void Border_Radius(this VisualElement element, params StyleKeyword[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword[]	args	These value will be applied to the element.

Border_Radius_Bottom_Left(VisualElement, Single, LengthUnit)

This will set the element's style borderBottomLeftRadius.

Declaration

```
public static void Border_Radius_Bottom_Left(this VisualElement element, float radius, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	radius	
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Border_Radius_Bottom_Left(VisualElement, StyleKeyword)

This will set the element's style borderBottomLeftRadius.

Declaration

```
public static void Border_Radius_Bottom_Left(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)

This will set the element's style borderBottomRightRadius.

Declaration

```
public static void Border_Radius_Bottom_Right(this VisualElement element, float radius, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	radius	
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Border_Radius_Bottom_Right(VisualElement, StyleKeyword)

This will set the element's style borderBottomRightRadius.

Declaration

```
public static void Border_Radius_Bottom_Right(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Border_Radius_Top_Left(VisualElement, Single, LengthUnit)

This will set the element's style borderTopLeftRadius.

Declaration

```
public static void Border_Radius_Top_Left(this VisualElement element, float radius, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	radius	
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Border_Radius_Top_Left(VisualElement, StyleKeyword)

This will set the element's style borderTopLeftRadius.

Declaration

```
public static void Border_Radius_Top_Left(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Border_Radius_Top_Right(VisualElement, Single, LengthUnit)

This will set the element's style borderTopRightRadius.

Declaration

```
public static void Border_Radius_Top_Right(this VisualElement element, float radius, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	radius	
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Border_Radius_Top_Right(VisualElement, StyleKeyword)

This will set the element's style borderTopRightRadius.

Declaration

```
public static void Border_Radius_Top_Right(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Border_Width(VisualElement, Single[])

This will set the element's style borderTopWidth, borderRightWidth, borderBottomWidth & borderLeftWidth.

Declaration

```
public static void Border_Width(this VisualElement element, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single[]	args	These value will be applied to the element.

Border_Width(VisualElement, StyleKeyword[])

This will set the element's style borderTopWidth, borderRightWidth, borderBottomWidth & borderLeftWidth.

Declaration

```
public static void Border_Width(this VisualElement element, params StyleKeyword[] args)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword[]	args	These value will be applied to the element.

Border_Width_Bottom(VisualElement, Single)

This will set the element's style borderBottomWidth.

Declaration

```
public static void Border_Width_Bottom(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.

Border_Width_Bottom(VisualElement, StyleKeyword)

This will set the element's style borderBottomWidth.

Declaration

```
public static void Border_Width_Bottom(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Border_Width_Left(VisualElement, Single)

This will set the element's style borderLeftWidth.

Declaration

```
public static void Border_Width_Left(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.

Border_Width_Left(VisualElement, StyleKeyword)

This will set the element's style `borderLeftWidth`.

Declaration

```
public static void Border_Width_Left(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Border_Width_Right(VisualElement, Single)

This will set the element's style `borderRightWidth`.

Declaration

```
public static void Border_Width_Right(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

TYPE	NAME	DESCRIPTION
System.Single	value	This value will be applied to the element.

Border_Width_Right(VisualElement, StyleKeyword)

This will set the element's style borderRightWidth.

Declaration

```
public static void Border_Width_Right(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Border_Width_Top(VisualElement, Single)

This will set the element's style borderTopWidth.

Declaration

```
public static void Border_Width_Top(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.

Border_Width_Top(VisualElement, StyleKeyword)

This will set the element's style borderTopWidth.

Declaration

```
public static void Border_Width_Top(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Color(VisualElement, Color)

This will set the element's style

Declaration

```
public static void Color(this VisualElement element, Color value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.Color	value	This value will be applied to the element.

Color(VisualElement, StyleKeyword)

This will set the element's style

Declaration

```
public static void Color(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Display_Hide(VisualElement)

This will set the element's style display

Declaration

```
public static void Display_Hide(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

Display_IsInvisible(VisualElement)

This will get the element's style display

Declaration

```
public static bool Display_IsInvisible(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be checked.

Returns

TYPE	DESCRIPTION
System.Boolean	return true if display is None.

Display_IsVisible(VisualElement)

This will get the element's style display

Declaration

```
public static bool Display_IsVisible(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be checked.

Returns

TYPE	DESCRIPTION
System.Boolean	return true if display is Flex.

Display_Show(VisualElement)

This will set the element's style display

Declaration

```
public static void Display_Show(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

Display_ToggleVisibility(VisualElement)

This will set the element's style display

Declaration

```
public static void Display_ToggleVisibility(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

Flex_Basis(VisualElement, Single)

This will set the element's style flexBasis

Declaration

```
public static void Flex_Basis(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.

Flex_Basis(VisualElement, StyleKeyword)

This will set the element's style flexBasis

Declaration

```
public static void Flex_Basis(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Flex_Direction(VisualElement, FlexDirection)

This will set the element's style flexDirection

Declaration

```
public static void Flex_Direction(this VisualElement element, FlexDirection value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.FlexDirection	value	This value will be applied to the element.

Flex_Direction(VisualElement, StyleKeyword)

This will set the element's style flexDirection

Declaration

```
public static void Flex_Direction(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Flex_Grow(VisualElement, Single)

This will set the element's style flexGrow

Declaration

```
public static void Flex_Grow(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.

Flex_Grow(VisualElement, StyleKeyword)

This will set the element's style flexGrow

Declaration

```
public static void Flex_Grow(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Flex_Shrink(VisualElement, Single)

This will set the element's style flexShrink

Declaration

```
public static void Flex_Shrink(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.

Flex_Shrink(VisualElement, StyleKeyword)

This will set the element's style flexShrink

Declaration

```
public static void Flex_Shrink(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Flex_Wrap(VisualElement, StyleKeyword)

This will set the element's style flexWrap

Declaration

```
public static void Flex_Wrap(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Flex_Wrap(VisualElement, Wrap)

This will set the element's style flexWrap

Declaration

```
public static void Flex_Wrap(this VisualElement element, Wrap value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.Wrap	value	This value will be applied to the element.

Font(VisualElement, Font)

This will set the element's style unity Font Definition.

Declaration

```
public static void Font(this VisualElement element, Font value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.Font	value	This value will be applied to the element.

Font(VisualElement, FontAsset)

This will set the element's style unity Font Definition.

Declaration

```
public static void Font(this VisualElement element, FontAsset value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.TextCore.Text.FontAsset	value	This value will be applied to the element.

Font_Size(VisualElement, Single)

This will set the element's style font size.

Declaration

```
public static void Font_Size(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.

Font_Size(VisualElement, StyleKeyword)

This will set the element's style font size.

Declaration

```
public static void Font_Size(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

GetBorder_Height(VisualElement)

This will get the element's style borderTopWidth & borderBottomWidth.

Declaration

```
public static float GetBorder_Height(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

Returns

TYPE	DESCRIPTION
System.Single	The return is the hight of the border.

GetBorder_Size(VisualElement)

This will get the element's style borderTopWidth, borderRightWidth, borderBottomWidth & borderLeftWidth.

Declaration

```
public static Vector2 GetBorder_Size(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	The return is a Vector2 of the size of the border.

GetBorder_Width(VisualElement)

This will get the element's style borderLeftWidth & borderRightWidth.

Declaration

```
public static float GetBorder_Width(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

Returns

TYPE	DESCRIPTION
System.Single	The return is the width of the border.

GetDisplay(VisualElement)

This will get this element's visible.

Declaration

```
public static DisplayStyle GetDisplay(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's visible changed.

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.DisplayStyle	

GetMargin_Height(VisualElement)

This will get the element's style marginTop & marginBottom.

Declaration

```
public static float GetMargin_Height(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

Returns

TYPE	DESCRIPTION
System.Single	

GetMargin_Size(VisualElement)

This will get the element's style marginTop, marginRight, marginBottom & marginLeft.

Declaration

```
public static Vector2 GetMargin_Size(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	

GetMargin_Width(VisualElement)

This will get the element's style marginLeft & marginRight.

Declaration

```
public static float GetMargin_Width(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

Returns

TYPE	DESCRIPTION
System.Single	

GetPadding_Height(VisualElement)

This will get the element's style paddingTop & paddingBottom.

Declaration

```
public static float GetPadding_Height(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

Returns

TYPE	DESCRIPTION
System.Single	

GetPadding_Size(VisualElement)

This will get the element's style paddingTop, paddingRight, paddingBottom & paddingLeft.

Declaration

```
public static Vector2 GetPadding_Size(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	The return is a Vector2 of the padding width & Height.

GetPadding_Width(VisualElement)

This will get the element's style paddingLeft & paddingRight.

Declaration

```
public static float GetPadding_Width(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

Returns

TYPE	DESCRIPTION
System.Single	

GetRadius_Bottom(VisualElement)

This will get the element's style borderBottomLeftRadius & borderBottomRightRadius.

Declaration

```
public static float GetRadius_Bottom(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

Returns

TYPE	DESCRIPTION
System.Single	The return is the size of the border radius Bottom.

GetRadius_Size(VisualElement)

This will get the element's style borderTopLeftRadius, borderTopRightRadius, borderBottomLeftRadius & borderBottomRightRadius.

Declaration

```
public static Vector2 GetRadius_Size(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	The return is a Vector2 the size of the border radius.

GetRadius_Top(VisualElement)

This will get the element's style borderTopLeftRadius & borderTopRightRadius.

Declaration

```
public static float GetRadius_Top(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

Returns

TYPE	DESCRIPTION
System.Single	The return is the size of the border radius Top.

GetSize(VisualElement, Boolean, Boolean)

Declaration

```
public static Vector2 GetSize(this VisualElement element, bool margin = true, bool padding = true)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	
System.Boolean	margin	
System.Boolean	padding	

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	

GetSize_Max(VisualElement, Boolean, Boolean)

This will Get the element's style maxWidth, maxHeight, margin, & padding.

Declaration

```
public static Vector2 GetSize_Max(this VisualElement element, bool margin = true, bool padding = true)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Boolean	margin	If true then add margin size to the element.
System.Boolean	padding	If true then add padding size to the element.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	The return is the max size of the element, with margin & padding if true.

GetSize_Min(VisualElement, Boolean, Boolean)

This will set the element's style minWidth, minHeight, margin, & padding.

Declaration

```
public static Vector2 GetSize_Min(this VisualElement element, bool margin = true, bool padding = true)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Boolean	margin	If true then add margin size to the element.
System.Boolean	padding	If true then add padding size to the element.

Returns

TYPE	DESCRIPTION
UnityEngine.Vector2	The return is the max size of the element, with margin & padding if true.

Margin(VisualElement, Single[])

This will set the element's style marginTop, marginRight, marginBottom & marginLeft.

Declaration

```
public static void Margin(this VisualElement element, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single[]	args	These value will be applied to the element.

Margin(VisualElement, LengthUnit, Single[])

This will set the element's style

Declaration

```
public static void Margin(this VisualElement element, LengthUnit unit, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.
System.Single[]	args	These value will be applied to the element.

Margin(VisualElement, LengthUnit[], Single[])

This will set the element's style

Declaration

```
public static void Margin(this VisualElement element, LengthUnit[] unit, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.LengthUnit[]	unit	This unit will be applied to the element.
System.Single[]	args	These value will be applied to the element.

Margin(VisualElement, StyleKeyword[])

This will set the element's style

Declaration

```
public static void Margin(this VisualElement element, params StyleKeyword[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.StyleKeyword[]	args	These value will be applied to the element.

Margin_Bottom(VisualElement, Single, LengthUnit)

This will set the element's style marginBottom.

Declaration

```
public static void Margin_Bottom(this VisualElement element, float value, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Margin_Bottom(VisualElement, StyleKeyword)

This will set the element's style marginBottom.

Declaration

```
public static void Margin_Bottom(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Margin_Left(VisualElement, Single, LengthUnit)

This will set the element's style marginLeft.

Declaration

```
public static void Margin_Left(this VisualElement element, float value, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Margin_Left(VisualElement, StyleKeyword)

This will set the element's style marginLeft.

Declaration

```
public static void Margin_Left(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Margin_Right(VisualElement, Single, LengthUnit)

This will set the element's style marginRight.

Declaration

```
public static void Margin_Right(this VisualElement element, float value, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Margin_Right(VisualElement, StyleKeyword)

This will set the element's style marginRight.

Declaration

```
public static void Margin_Right(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Margin_Top(VisualElement, Single, LengthUnit)

This will set the element's style marginTop.

Declaration

```
public static void Margin_Top(this VisualElement element, float value, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Margin_Top(VisualElement, StyleKeyword)

This will set the element's style marginTop.

Declaration

```
public static void Margin_Top(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Overflow(VisualElement, Overflow)

This will set the element's style overflow.

Declaration

```
public static void Overflow(this VisualElement element, Overflow value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.Overflow	value	This value will be applied to the element.

Overflow(VisualElement, StyleKeyword)

This will set the element's style overflow.

Declaration

```
public static void Overflow(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Padding(VisualElement, Single[])

This will set the element's style paddingTop, paddingRight, paddingBottom & paddingLeft.

Declaration

```
public static void Padding(this VisualElement element, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single[]	args	These value will be applied to the element.

Padding(VisualElement, LengthUnit, Single[])

This will set the element's style paddingTop, paddingRight, paddingBottom & paddingLeft.

Declaration

```
public static void Padding(this VisualElement element, LengthUnit unit = LengthUnit.Pixel, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.
System.Single[]	args	These value will be applied to the element.

Padding(VisualElement, LengthUnit[], Single[])

This will set the element's style paddingTop, paddingRight, paddingBottom & paddingLeft.

Declaration

```
public static void Padding(this VisualElement element, LengthUnit[] unit, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.LengthUnit[]	unit	This unit will be applied to the element.
System.Single[]	args	These value will be applied to the element.

Padding(VisualElement, StyleKeyword[])

This will set the element's style paddingTop, paddingRight, paddingBottom & paddingLeft.

Declaration

```
public static void Padding(this VisualElement element, params StyleKeyword[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword[]	args	These value will be applied to the element.

Padding_Bottom(VisualElement, Single, LengthUnit)

This will set the element's style paddingBottom.

Declaration

```
public static void Padding_Bottom(this VisualElement element, float value, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Padding_Bottom(VisualElement, StyleKeyword)

This will set the element's style paddingBottom.

Declaration

```
public static void Padding_Bottom(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Padding_Left(VisualElement, Single, LengthUnit)

This will set the element's style paddingLeft.

Declaration

```
public static void Padding_Left(this VisualElement element, float value, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Padding_Left(VisualElement, StyleKeyword)

This will set the element's style paddingLeft.

Declaration

```
public static void Padding_Left(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Padding_Right(VisualElement, Single, LengthUnit)

This will set the element's style paddingRight.

Declaration

```
public static void Padding_Right(this VisualElement element, float value, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Padding_Right(VisualElement, StyleKeyword)

This will set the element's style paddingRight.

Declaration

```
public static void Padding_Right(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Padding_Top(VisualElement, Single, LengthUnit)

This will set the element's style paddingTop.

Declaration

```
public static void Padding_Top(this VisualElement element, float value, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Padding_Top(VisualElement, StyleKeyword)

Declaration

```
public static void Padding_Top(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Position_Bottom_Left(VisualElement, Single, Single)

This will set the element's style Bottom & Left.

Declaration

```
public static void Position_Bottom_Left(this VisualElement element, float x, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	x	
System.Single	y	

Position_Bottom_Left(VisualElement, Vector2)

This will set the element's style Bottom & Left.

Declaration

```
public static void Position_Bottom_Left(this VisualElement element, Vector2 value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.Vector2	value	This value will be applied to the element.

Position_Bottom_Right(VisualElement, Single, Single)

This will set the element's style Bottom & Right.

Declaration

```
public static void Position_Bottom_Right(this VisualElement element, float x, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	x	This value will be applied to the element.
System.Single	y	This value will be applied to the element.

Position_Bottom_Right(VisualElement, Vector2)

This will set the element's style Bottom & Right.

Declaration

```
public static void Position_Bottom_Right(this VisualElement element, Vector2 value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.Vector2	value	This value will be applied to the element.

Position_Top_Left(VisualElement, Single, Single)

This will set the element's style Top & Left.

Declaration

```
public static void Position_Top_Left(this VisualElement element, float x, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	x	
System.Single	y	

Position_Top_Left(VisualElement, Vector2)

This will set the element's style Top & Left.

Declaration

```
public static void Position_Top_Left(this VisualElement element, Vector2 value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.Vector2	value	This value will be applied to the element.

Position_Top_Right(VisualElement, Single, Single)

This will set the element's style Top & Right.

Declaration

```
public static void Position_Top_Right(this VisualElement element, float x, float y)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	x	
System.Single	y	

Position_Top_Right(VisualElement, Vector2)

This will set the element's style Top & Right.

Declaration

```
public static void Position_Top_Right(this VisualElement element, Vector2 value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.Vector2	value	This value will be applied to the element.

SetDisplay(VisualElement, DisplayStyle)

This will set the visible from value.

Declaration

```
public static void SetDisplay(this VisualElement element, DisplayStyle value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	The element that will have it's visible changed.
UnityEngine.UIElements.DisplayStyle	value	The DisplayStyle that will change value.

Size(VisualElement, Single[])

This will set the element's style width & height.

Declaration

```
public static void Size(this VisualElement element, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single[]	args	These value will be applied to the element.

Size(VisualElement, LengthUnit, Single[])

This will set the element's style width & height.

Declaration

```
public static void Size(this VisualElement element, LengthUnit unit = LengthUnit.Pixel, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.
System.Single[]	args	These value will be applied to the element.

Size(VisualElement, LengthUnit, LengthUnit, Single[])

This will set the element's style width & height.

Declaration

```
public static void Size(this VisualElement element, LengthUnit unitWidth, LengthUnit unitHeight, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.LengthUnit	unitWidth	This unit will be applied to the element.
UnityEngine.UIElements.LengthUnit	unitHeight	This unit will be applied to the element.
System.Single[]	args	These value will be applied to the element.

Size(VisualElement, StyleKeyword)

This will set the element's style width & height.

Declaration

```
public static void Size(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Size(VisualElement, Vector2, LengthUnit[])

This will set the element's style width & height.

Declaration

```
public static void Size(this VisualElement element, Vector2 value, params LengthUnit[] unit)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.Vector2	value	This value will be applied to the element.
UnityEngine.UIElements.LengthUnit[]	unit	This unit will be applied to the element.

Size_Height(VisualElement, Single, LengthUnit)

This will set the element's style height.

Declaration

```
public static void Size_Height(this VisualElement element, float value, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Size_Height(VisualElement, StyleKeyword)

This will set the element's style height.

Declaration

```
public static void Size_Height(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Size_Height_Max(VisualElement, Single, LengthUnit)

This will set the element's style maxHeight.

Declaration

```
public static void Size_Height_Max(this VisualElement element, float value, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Size_Height_Max(VisualElement, StyleKeyword)

This will set the element's style maxHeight.

Declaration

```
public static void Size_Height_Max(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Size_Height_Min(VisualElement, Single, LengthUnit)

This will set the element's style minHeight

Declaration

```
public static void Size_Height_Min(this VisualElement element, float value, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Size_Height_Min(VisualElement, StyleKeyword)

This will set the element's style minHeight

Declaration

```
public static void Size_Height_Min(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Size_Max(VisualElement, Single[])

This will set the element's style maxWidth & maxHeight.

Declaration

```
public static void Size_Max(this VisualElement element, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single[]	args	These value will be applied to the element.

Size_Max(VisualElement, LengthUnit, Single[])

This will set the element's style maxWidth & maxHeight.

Declaration

```
public static void Size_Max(this VisualElement element, LengthUnit unit = LengthUnit.Pixel, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.
System.Single[]	args	These value will be applied to the element.

Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])

This will set the element's style maxWidth & maxHeight.

Declaration

```
public static void Size_Max(this VisualElement element, LengthUnit unitWidth, LengthUnit unitHeight, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.LengthUnit	unitWidth	This unit will be applied to the element.
UnityEngine.UIElements.LengthUnit	unitHeight	This unit will be applied to the element.
System.Single[]	args	These value will be applied to the element.

Size_Max(VisualElement, StyleKeyword)

This will set the element's style maxWidth & maxHeight.

Declaration

```
public static void Size_Max(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Size_Max(VisualElement, Vector2, LengthUnit[])

This will set the element's style maxWidth & maxHeight.

Declaration

```
public static void Size_Max(this VisualElement element, Vector2 value, params LengthUnit[] unit)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.Vector2	value	This value will be applied to the element.
UnityEngine.UIElements.LengthUnit[]	unit	This unit will be applied to the element.

Size_Min(VisualElement, Single[])

This will set the element's style minWidth & minHeight.

Declaration

```
public static void Size_Min(this VisualElement element, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single[]	args	These value will be applied to the element.

Size_Min(VisualElement, LengthUnit, Single[])

This will set the element's style minWidth & minHeight.

Declaration

```
public static void Size_Min(this VisualElement element, LengthUnit unit = LengthUnit.Pixel, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

TYPE	NAME	DESCRIPTION
System.Single[]	args	These value will be applied to the element.

Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])

This will set the element's style minWidth & minHeight.

Declaration

```
public static void Size_Min(this VisualElement element, LengthUnit unitWidth, LengthUnit unitHeight, params float[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.LengthUnit	unitWidth	This unit will be applied to the element.
UnityEngine.UIElements.LengthUnit	unitHeight	This unit will be applied to the element.
System.Single[]	args	These value will be applied to the element.

Size_Min(VisualElement, StyleKeyword)

This will set the element's style minWidth & minHeight.

Declaration

```
public static void Size_Min(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Size_Min(VisualElement, Vector2, LengthUnit[])

This will set the element's style minWidth & minHeight.

Declaration

```
public static void Size_Min(this VisualElement element, Vector2 value, params LengthUnit[] unit)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.Vector2	value	This value will be applied to the element.
UnityEngine.UIElements.LengthUnit[]	unit	This unit will be applied to the element.

Size_Width(VisualElement, Single, LengthUnit)

This will set the element's style width.

Declaration

```
public static void Size_Width(this VisualElement element, float value, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Size_Width(VisualElement, StyleKeyword)

This will set the element's style width.

Declaration

```
public static void Size_Width(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Size_Width_Max(VisualElement, Single, LengthUnit)

This will set the element's style maxWidth.

Declaration

```
public static void Size_Width_Max(this VisualElement element, float value, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Size_Width_Max(VisualElement, StyleKeyword)

This will set the element's style maxWidth.

Declaration

```
public static void Size_Width_Max(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Size_Width_Min(VisualElement, Single, LengthUnit)

This will set the element's style minWidth

Declaration

```
public static void Size_Width_Min(this VisualElement element, float value, LengthUnit unit = LengthUnit.Pixel)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.
UnityEngine.UIElements.LengthUnit	unit	This unit will be applied to the element.

Size_Width_Min(VisualElement, StyleKeyword)

This will set the element's style minWidth

Declaration

```
public static void Size_Width_Min(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

StylePosition(VisualElement, Position)

This will set the element's style position.

Declaration

```
public static void StylePosition(this VisualElement element, Position value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.Position	value	This value will be applied to the element.

StylePosition(VisualElement, StyleKeyword)

This will set the element's style position.

Declaration

```
public static void StylePosition(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

StylePosition_Bottom(VisualElement, Single)

This will set the element's style bottom.

Declaration

```
public static void StylePosition_Bottom(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.

StylePosition_Bottom(VisualElement, StyleKeyword)

This will set the element's style bottom.

Declaration

```
public static void StylePosition_Bottom(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

StylePosition_Left(VisualElement, Single)

This will set the element's style left.

Declaration

```
public static void StylePosition_Left(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.

StylePosition_Left(VisualElement, StyleKeyword)

This will set the element's style left.

Declaration

```
public static void StylePosition_Left(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

StylePosition_Right(VisualElement, Single)

This will set the element's style right.

Declaration

```
public static void StylePosition_Right(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.

StylePosition_Right(VisualElement, StyleKeyword)

This will set the element's style right.

Declaration

```
public static void StylePosition_Right(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

StylePosition_Toggle(VisualElement)

This will set the element's style position.

Declaration

```
public static void StylePosition_Toggle(this VisualElement element)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.

StylePosition_Top(VisualElement, Single)

This will set the element's style top.

Declaration

```
public static void StylePosition_Top(this VisualElement element, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
System.Single	value	This value will be applied to the element.

StylePosition_Top(VisualElement, StyleKeyword)

This will set the element's style top.

Declaration

```
public static void StylePosition_Top(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

Text_Position(VisualElement, TextAnchor)

This will set the element's style unity Text Align.

Declaration

```
public static void Text_Position(this VisualElement element, TextAnchor value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.TextAnchor	value	This value will be applied to the element.

White_Space(VisualElement, StyleKeyword)

This will set the element's style white Space

Declaration

```
public static void White_Space(this VisualElement element, StyleKeyword value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.StyleKeyword	value	This value will be applied to the element.

White_Space(VisualElement, WhiteSpace)

This will set the element's style white Space

Declaration

```
public static void White_Space(this VisualElement element, WhiteSpace value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	element	This is the element that will be changed.
UnityEngine.UIElements.WhiteSpace	value	This value will be applied to the element.

Namespace UtilitySystem.Script

Classes

Assets

This static class contains boilerplate method(s) relating to AssetDatabase.

CreateElements

This static class will allow the creation of UI Toolkit elements.

Movement

This class is drop an go movement script that will follow the player object. useful for testing.

RichText

This static class will simplify setting up rich text for "Text Mesh Pro" or "UI Toolkit"

Class Assets

This static class contains boilerplate method(s) relating to AssetDatabase.

Inheritance

System.Object

Assets

Namespace: [UtilitySystem.Script](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public static class Assets
```

Class CreateElements

This static class will allow the creation of UI Toolkit elements.

Inheritance

System.Object
CreateElements

Namespace: [UtilitySystem.Script](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public static class CreateElements
```

Methods

CreateElement_Button(VisualElement, Action)

This will create a Button and add to the element that invokes this.

Declaration

```
public static Button CreateElement_Button(this VisualElement parent, Action action)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.Action	action	This is the action when the Button is clicked.

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	The return is the Button that is created.

CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])

This will create a ContextMenuElement and add to the element that invokes this.

Declaration

```
public static ContextMenuElement CreateElement_ContextMenu(this VisualElement parent, params ContextMenuItemData[] data)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
ContextMenuItemData[]	data	These params are the data of the Context Menu.

Returns

TYPE	DESCRIPTION
ContextMenuElement	The return is the ContextMenuElement that is created.

CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)

This will create a ContextMenuElement and add to the element that invokes this.

Declaration

```
public static ContextMenuElement CreateElement_ContextMenu(this VisualElement parent,
List<ContextMenuItemData> data)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.Collections.Generic.List< ContextMenuItemData >	data	These are the data of the Context Menu.

Returns

TYPE	DESCRIPTION
ContextMenuElement	The return is the ContextMenuElement that is created.

CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)

This will create a ContextMenuItemElement and add to the element that invokes this.

Declaration

```
public static ContextMenuItemElement CreateElement_ContextMenu_Item(this VisualElement parent, ContextMenuItemData data)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
ContextMenuItemData	data	This is the data of the Context Menu Item.

Returns

TYPE	DESCRIPTION
ContextMenuItemElement	The return is the ContextMenuItemElement that is created.

CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)

This will create a ContextMenuItemButtonElement and add to the element that invokes this.

Declaration

```
public static ContextMenuItemButtonElement CreateElement_ContextMenu_Item_Button(this VisualElement parent, ContextMenuItemButtonData data)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
ContextMenuItemButtonData	data	This is the data of the Context Menu button.

Returns

TYPE	DESCRIPTION
ContextMenuItemButtonElement	The return is the ContextMenuItemButtonElement that is created.

CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)

This will create a ContextMenuItemParentElement and add to the element that invokes this.

Declaration

```
public static ContextMenuItemParentElement CreateElement_ContextMenu_Item_Parent(this VisualElement parent, ContextMenuItemParentData data)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
ContextMenuItemParentData	data	This is the data of the Context Menu Parent.

Returns

TYPE	DESCRIPTION
ContextMenuItemParentElement	The return is the ContextMenuItemParentElement that is created.

CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)

This will create a ContextMenuItemElement and add to the element that invokes this.

Declaration

```
public static ContextMenuItemElement CreateElement_ContextMenuItem(this VisualElement parent, ContextMenuItemData data)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
ContextMenuItemData	data	This is the data of the Context Menu.

Returns

TYPE	DESCRIPTION
ContextMenuItemElement	The return is the ContextMenuItemElement that is created.

CreateElement_DialogBox(VisualElement, String, String, Boolean)

This will create a DialogBoxElement and add to the element that invokes this.

Declaration

```
public static DialogBoxElement CreateElement_DialogBox(this VisualElement parent, string UID, string
titleText, bool isDisplay = false)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.String	UID	This is the Unique ID of the DialogBox.
System.String	titleText	This is the title of the DialogBox.
System.Boolean	isDisplay	

Returns

TYPE	DESCRIPTION
DialogBoxElement	The return is the DialogBoxElement that is created.

CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)

This will create a DialogBoxElement and add to the element that invokes this.

Declaration

```
public static DialogBoxElement CreateElement_DialogBox(this VisualElement parent, string UID, string
titleText, VisualElement content, bool isDisplay = false)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.String	UID	This is the Unique ID of the DialogBox.

TYPE	NAME	DESCRIPTION
System.String	titleText	This is the title of the DialogBox.
UnityEngine.UIElements.VisualElement	content	
System.Boolean	isDisplay	

Returns

TYPE	DESCRIPTION
DialogBoxElement	The return is the DialogBoxElement that is created.

CreateElement_Dropdown(VisualElement, String, List<String>, Int32)

This will create a DropdownField and add to the element that invokes this.

Declaration

```
public static DropdownField CreateElement_Dropdown(this VisualElement parent, string label = null, List<string> choices = null, int defaultIndex = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.String	label	This is the label of the DropdownField.
System.Collections.Generic.List<System.String>	choices	This is the choices of the DropdownField.
System.Int32	defaultIndex	This is the defaultIndex of the DropdownField.

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.DropdownField	The return is the DropdownField that is created.

CreateElement_Foldout(VisualElement, String, Boolean)

This will create a Foldout and add to the element that invokes this.

Declaration

```
public static Foldout CreateElement_Foldout(this VisualElement parent, string label = null, bool isOpen = false)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.String	label	This is the label of the Foldout.
System.Boolean	isOpen	This is the boolean that opens the Foldout.

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.Foldout	The return is the Foldout that is created.

CreateElement_Label(VisualElement, String, Boolean)

This will create a Label and add to the element that invokes this.

Declaration

```
public static Label CreateElement_Label(this VisualElement parent, string text, bool enableRichText = true)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.String	text	This is the text of the Label.
System.Boolean	enableRichText	This is the boolean that will enable Rich Text for the Label.

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.Label	The return is the Label that is created.

CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)

This will create a ListView and add to the element that invokes this.

Declaration

```
public static ListView CreateElement_ListView<TValue>(this VisualElement parent, List<TValue> items, float itemHeight = -1F, SelectionType type = SelectionType.Single)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.Collections.Generic.List<TValue>	items	This is the items of the ListView.
System.Single	itemHeight	This is the height of the item in the list in pixel of the ListView.
UnityEngine.UIElements.SelectionType	type	This is the type of selection for the ListView.

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.ListView	The return is the ListView that is created.

Type Parameters

NAME	DESCRIPTION
TValue	

CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)

This will create a ProgressBar and add to the element that invokes this.

Declaration

```
public static ProgressBar CreateElement_Progress_Bar(this VisualElement parent, string title, float lowValue, float highValue, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.String	title	This is the title of the ProgressBar.
System.Single	lowValue	This is the lowValue of the ProgressBar.
System.Single	highValue	This is the highValue of the ProgressBar.
System.Single	value	This is the value of the ProgressBar.

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.ProgressBar	The return is the ProgressBar that is created.

CreateElement_Radio(VisualElement, Boolean, String, Boolean)

This will create a RadioButton and add to the element that invokes this.

Declaration

```
public static RadioButton CreateElement_Radio(this VisualElement parent, bool value, string label = null, bool isSelected = false)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.Boolean	value	
System.String	label	This is the label of the RadioButton.

TYPE	NAME	DESCRIPTION
System.Boolean	isSelected	If true the radio button will be selected.

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.RadioButton	The return is the RadioButton that is created.

CreateElement_Radio_Group(VisualElement, String, String[])

This will create a RadioButtonGroup and add to the element that invokes this.

Declaration

```
public static RadioButtonGroup CreateElement_Radio_Group(this VisualElement parent, string label = null,
params string[] choices)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.String	label	This is the label of the RadioButtonGroup.
System.String[]	choices	This is the choices of the RadioButtonGroup.

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.RadioButtonGroup	The return is the RadioButtonGroup that is created.

CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])

This will create a ScrollView and add to the element that invokes this.

Declaration

```
public static ScrollView CreateElement_ScrollView(this VisualElement parent, ScrollViewMode scrollViewMode,
params VisualElement[] children)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
UnityEngine.UIElements.ScrollViewMode	scrollViewMode	This is the scroll view of the ScrollView.
UnityEngine.UIElements.VisualElement[]	children	This is params the children of the ScrollView.

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.ScrollView	The return is the ScrollView that is created.

CreateElement_Slider(VisualElement, Single, Single, String)

This will create a Slider and add to the element that invokes this.

Declaration

```
public static Slider CreateElement_Slider(this VisualElement parent, float minAmount, float MaxAmount, string label = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.Single	minAmount	This is the minAmount of the Slider.
System.Single	MaxAmount	This is the MaxAmount of the Slider.
System.String	label	This is the label of the Slider.

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.Slider	The return is the Slider that is created.

CreateElement_Slider_Int(VisualElement, Int32, Int32, String)

This will create a SliderInt and add to the element that invokes this.

Declaration

```
public static SliderInt CreateElement_Slider_Int(this VisualElement parent, int minAmount, int MaxAmount, string label = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.Int32	minAmount	This is the minAmount of the SliderInt.
System.Int32	MaxAmount	This is the MaxAmount of the SliderInt.
System.String	label	This is the label of the SliderInt.

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.SliderInt	The return is the SliderInt that is created.

CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)

This will create a MinMaxSlider and add to the element that invokes this.

Declaration

```
public static MinMaxSlider CreateElement_Slider_Min_Max(this VisualElement parent, string label = null, float minValue = 0F, float maxValue = 10F, float minLimit = -3.40282347E+38F, float maxLimit = 3.40282347E+38F)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.String	label	This is the label of the MinMaxSlider.
System.Single	minValue	This is the minValue of the MinMaxSlider.
System.Single	maxValue	This is the maxValue of the MinMaxSlider.
System.Single	minLimit	This is the minLimit of the MinMaxSlider.
System.Single	maxLimit	This is the maxLimit of the MinMaxSlider.

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.MinMaxSlider	The return is the MinMaxSlider that is created.

CreateElement_Sprite(VisualElement, Sprite)

This will create a SpriteElement and add to the element that invokes this.

Declaration

```
public static SpriteElement CreateElement_Sprite(this VisualElement parent, Sprite sprite)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
UnityEngine.Sprite	sprite	This is the sprite of the element.

Returns

TYPE	DESCRIPTION
SpriteElement	The return is the SpriteElement that is created.

CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)

This will create a TextField and add to the element that invokes this.

Declaration

```
public static TextField CreateElement_TextField(this VisualElement parent, string label = null, int maxLength = 0, bool multiline = false, bool isPasswordField = false, char maskChar = '*')
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.String	label	This is the label of the TextField.
System.Int32	maxLength	This is the maxLength of the TextField.
System.Boolean	multiline	This will determine of the TextField.
System.Boolean	isPasswordField	This is the boolean for passwords.
System.Char	maskChar	This is the character that will be used for the password.

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.TextField	The return is the TextField that is created.

CreateElement_Toggle(VisualElement, String)

This will create a Toggle and add to the element that invokes this.

Declaration

```
public static Toggle CreateElement_Toggle(this VisualElement parent, string label)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
System.String	label	This is the label of the Toggle.

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.Toggle	The return is the Toggle that is created.

CreateElement_Tooltip(VisualElement, Font, Int32)

This will create a TooltipElement and add to the element that invokes this.

Declaration

```
public static TooltipElement CreateElement_Tooltip(this VisualElement parent, Font font = null, int textSize = 0)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.
UnityEngine.Font	font	This is the font of the tooltip.
System.Int32	textSize	This is the size of the tooltip.

Returns

TYPE	DESCRIPTION
TooltipElement	The return is the TooltipElement that is created.

CreateElement_VisualElement(VisualElement)

This will create a VisualElement and add to the element that invokes this.

Declaration

```
public static VisualElement CreateElement_VisualElement(this VisualElement parent)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	parent	This param is the element that invokes the method.

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.VisualElement	The return is the VisualElement that is created.

Class Movement

This class is drop an go movement script that will follow the player object. useful for testing.

Inheritance

System.Object
UnityEngine.Object
UnityEngine.Component
UnityEngine.Behaviour
UnityEngine.MonoBehaviour
Movement

Namespace: [UtilitySystem.Script](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class Movement : MonoBehaviour
```

Fields

Instance

This is a singleton instance.

Declaration

```
public static Movement Instance
```

Field Value

TYPE	DESCRIPTION
Movement	

Extension Methods

- [RichText.UpperCaseText\(Object\)](#)
- [RichText.LowerCaseText\(Object\)](#)
- [RichText.BoldText\(Object\)](#)
- [RichText.ItalicText\(Object\)](#)
- [RichText.Text\(Object\)](#)
- [RichText.UnderLineText\(Object\)](#)
- [RichText.StrikeText\(Object\)](#)
- [RichText.SupText\(Object\)](#)
- [RichText.SubText\(Object\)](#)
- [RichText.PositionText\(Object, Single\)](#)
- [RichText.SizeText\(Object, Int32\)](#)
- [RichText.ColoredText\(Object, String\)](#)
- [RichText.ColoredText\(Object, Color\)](#)
- [RichText.BooleanText\(Object, Boolean\)](#)

RichText.BooleanText(Object, Boolean, Color)

Class RichText

This static class will simplify setting up rich text for "Text Mesh Pro" or "UI ToolKit"

Inheritance

System.Object
RichText

Namespace: [UtilitySystem.Script](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public static class RichText
```

Properties

NewLine

This will take an object and create a new line string.

Declaration

```
public static string NewLine { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Tab

This will take an object and create a tab gap string.

Declaration

```
public static string Tab { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

BoldText(Object)

This will take an object and create a string that is bold.

Declaration

```
public static string BoldText(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This will return a string that is set to bold.

BooleanText(Boolean)

This will take an object and create a string that is either colored green for true or red for false.

Declaration

```
public static string BooleanText(this bool arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This will return a string that is either colored green for true or red for false.

BooleanText(Object, Boolean)

This will take an object and create a string that is either colored green for true or red for false.

Declaration

```
public static string BooleanText(this object arg, bool check)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.
System.Boolean	check	This is the boolean that will determine if true or false.

Returns

TYPE	DESCRIPTION
System.String	This will return a string that is either colored green for true or red for false.

BooleanText(Object, Boolean, Color)

This will take an object and create a string that is either colored green for true or red for false.

Declaration

```
public static string BooleanText(this object arg, bool check, Color color)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.
System.Boolean	check	This is the boolean that will determine if true or false.
UnityEngine.Color	color	if true set the text to the color else return without color.

Returns

TYPE	DESCRIPTION
System.String	This will return a string that is either colored green for true or red for false.

ColoredText(Object, String)

This will take an object and create a string that will color the text.

Declaration

```
public static string ColoredText(this object arg, string color)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string. This is the object that will be passed in and will be added into a string.
System.String	color	this is the color that will be used to color the text.

Returns

TYPE	DESCRIPTION
System.String	This will return a string that is colored from the color param.

ColoredText(Object, Color)

This will take an object and create a string that will color the text.

Declaration

```
public static string ColoredText(this object arg, Color color)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.
UnityEngine.Color	color	this is the color that will be used to color the text.

Returns

TYPE	DESCRIPTION
System.String	This will return a string that is colored from the color param.

ItalicText(Object)

This will take an object and create a string that is italic.

Declaration

```
public static string ItalicText(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This will return a string that is given an italic lean.

LowerCaseText(Object)

This will return a string to an lowercase

Declaration

```
public static string LowerCaseText(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This will return a string that is all lowercase.

PositionText(Object, Single)

This will take an object and create a string that is a set position for the text on the line.

Declaration

```
public static string PositionText(this object arg, float pos)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.
System.Single	pos	This is a float that will be used to set the position of the text in the line.

Returns

TYPE	DESCRIPTION
System.String	This will return a string that is set at a position.

SetText(List<Func<String>>)

This will take an object and create a string that is either colored green for true or red for false.

Declaration

```
public static string SetText(List<Func<string>> messages)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<System.Func<System.String>>	messages	this is a list of Func strings that will be run and appended to a StringBuilder

Returns

TYPE	DESCRIPTION
System.String	This will return a string from the StringBuilder.

SizeText(Object, Int32)

This will take an object and create a string that is the size for the text.

Declaration

```
public static string SizeText(this object arg, int size)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.
System.Int32	size	This is an integer for the size of the text.

Returns

TYPE	DESCRIPTION
System.String	This will return a string that has it's size changed to the size param.

StrikeText(Object)

This will take an object and create a string that is strikethrough.

Declaration

```
public static string StrikeText(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This will return a string that has a line though it.

SubText(Object)

This will take an object and create a string that is lowered.

Declaration

```
public static string SubText(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This will return a string that is lower then other text.

SupText(Object)

This will take an object and create a string that is raised.

Declaration

```
public static string SupText(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This will return a string that is higher then other text .

Text(Object)

This will take an object and create a string.

Declaration

```
public static string Text(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This will return a string that is a generic text string.

UnderLineText(Object)

This will take an object and create a string that is underline.

Declaration

```
public static string UnderLineText(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This will return a string that is underlined.

UpperCaseText(Object)

This will return a string to an uppercase

Declaration

```
public static string UpperCaseText(this object arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	arg	This is the object that will be passed in and will be added into a string.

Returns

TYPE	DESCRIPTION
System.String	This will return a string that is all uppercase.

Namespace UtilitySystem.Tool

Classes

[Timer](#)

This class is a Timer that can be used to activate an action after an amount of time.

Class Timer

This class is a Timer that can be used to activate an action after an amount of time.

Inheritance

System.Object
Timer

Namespace: [UtilitySystem.Tool](#)
Assembly: Assembly-CSharp.dll

Syntax

```
[Serializable]  
public class Timer
```

Constructors

Timer(Action, Single, Boolean)

This is a constructor that will setup the timer.

Declaration

```
public Timer(Action action, float timer, bool isRunning = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action	action	the Action that will be run when the timer finishes.
System.Single	timer	the amount of time it will take to finish.
System.Boolean	isRunning	This param if true will set the timer to start or stop. (default true).

Properties

CurrentTime

This property will return the current time round up.

Declaration

```
public int CurrentTime { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Running

This will return true if the timer is running.

Declaration

```
public bool Running { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

ResetTimer()

This will reset the timer.

Declaration

```
public void ResetTimer()
```

StartTimer()

This will start the timer.

Declaration

```
public void StartTimer()
```

StopTimer()

This will stop the timer.

Declaration

```
public void StopTimer()
```

UpdateTimer()

This will update the timer.

Declaration

```
public void UpdateTimer()
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)

Namespace UtilitySystem.UI

Classes

[ContextMenuElement](#)

This class is the context menu for the runtime.

[ContextMenuElement.UxmlFactory](#)

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

[ContextMenuItemButtonElement](#)

This class is the context menu button for the runtime.

[ContextMenuItemButtonElement.UxmlFactory](#)

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

[ContextMenuItemElement](#)

This class is the context menu item for the runtime.

[ContextMenuItemElement.UxmlFactory](#)

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

[ContextMenuItemParentElement](#)

This class is the context sub-menu for the runtime.

[ContextMenuItemParentElement.UxmlFactory](#)

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

[DialogBoxElement](#)

This class is the dialog box that can be move around on the screen.

[DialogBoxElement.UxmlFactory](#)

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

[PopupBaseElement](#)

This abstract class that is the base of the popup elements.

SpriteElement

This Class is an VisualElement that is created for the Inventory Editor.

SpriteElement.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

ToolTipElement

This class will create a tooltip for your game.

ToolTipElement.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Class ContextMenuElement

This class is the context menu for the runtime.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
ContextMenuElement

Namespace: [UtilitySystem.UI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class ContextMenuElement : VisualElement, IEventHandler, ITransform, ITransitionAnimations, IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

ContextMenuElement()

This is a constructor and will create the context menu.

Declaration

```
public ContextMenuElement()
```

ContextMenuElement(ContextMenuItemData[])

This is a constructor and will create the context menu.

Declaration

```
public ContextMenuElement(params ContextMenuItemData[] data)
```

Parameters

TYPE	NAME	DESCRIPTION
ContextMenuItemData[]	data	these is the list of Context Menu Item Data

ContextMenuElement(List<ContextMenuItemData>)

This is a constructor and will create the context menu.

Declaration

```
public ContextMenuElement(List<ContextMenuItemData> data)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List< ContextMenuItemData >	data	this is the list of Context Menu Item Data

Methods

GenerateElements(List<ContextMenuItemData>)

This will generate "ContextMenu" elements for this menu

Declaration

```
public void GenerateElements(List<ContextMenuItemData> data)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List< ContextMenuItemData >	data	This is a list of ContextMenuItemData that will be used to create the menu items.

LineBreaker()

This will create a line breaker on the context menu.

Declaration

```
public VisualElement LineBreaker()
```

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.VisualElement	The return is the element that is created.

Style()

This will the style of the context menu.

Declaration

```
public virtual void Style()
```

Extension Methods

VisualElementStyleExtension.BackgroundColor(VisualElement, Color)
VisualElementStyleExtension.Background_Image(VisualElement, Background)
VisualElementStyleExtension.Background_Image(VisualElement, Texture2D)
VisualElementStyleExtension.Background_Image(VisualElement, Sprite)
VisualElementStyleExtension.Background_Image(VisualElement, StyleKeyword)
VisualElementStyleExtension.Color(VisualElement, Color)
VisualElementStyleExtension.Color(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Top(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Right(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Left(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color(VisualElement, Color[])
VisualElementStyleExtension.Border_Color(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetRadius_Top(VisualElement)
VisualElementStyleExtension.GetRadius_Bottom(VisualElement)
VisualElementStyleExtension.GetRadius_Size(VisualElement)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius(VisualElement, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetBorder_Width(VisualElement)
VisualElementStyleExtension.GetBorder_Height(VisualElement)
VisualElementStyleExtension.GetBorder_Size(VisualElement)
VisualElementStyleExtension.Border_Width_Top(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Right(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Left(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width(VisualElement, Single[])
VisualElementStyleExtension.Border_Width(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Border(VisualElement, Single[])
VisualElementStyleExtension.Border(VisualElement, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])

VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)
VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)

VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)
VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)

VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)
VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)
VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)
VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)
VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)
VisualElementExtension.SetTransform_Position(VisualElement, Vector2)
VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)

VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)
VisualElementExtension.GetElementSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)
VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)
CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)
CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32, Boolean)
CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)
CreateElements.CreateElement_Icon(VisualElement, InventoryItem)
CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)
CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class ContextMenuElement.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Inheritance

System.Object

UnityEngine.UIElements.UxmlFactory<[ContextMenuElement](#), [VisualElement.UxmlTraits](#)>

ContextMenuElement.UxmlFactory

Namespace: [UtilitySystem.UI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<ContextMenuElement, VisualElement.UxmlTraits>, IUxmlFactory
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)

Class ContextMenuItemButtonElement

This class is the context menu button for the runtime.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
[ContextMenuItemElement](#)
ContextMenuItemButtonElement

Namespace: [UtilitySystem.UI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class ContextMenuItemButtonElement : ContextMenuItemElement, IEventHandler, ITransform, ITransitionAnimations, IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

ContextMenuItemButtonElement()

This is a constructor and will create the context menu item.

Declaration

```
public ContextMenuItemButtonElement()
```

ContextMenuItemButtonElement(ContextMenuItemButtonData)

This is a constructor and will create the context menu item.

Declaration

```
public ContextMenuItemButtonElement(ContextMenuItemButtonData data)
```

Parameters

TYPE	NAME	DESCRIPTION
ContextMenuItemButtonData	data	this is the list of Context Menu Item Data

Fields

action

This is the action will be invoke when the item is clicked.

Declaration

```
public Action action
```

Field Value

TYPE	DESCRIPTION
System.Action	

Methods

GenerateElements(ContextMenuItemButtonData)

This will generate "ContextMenuitem" elements for this menu

Declaration

```
public void GenerateElements(ContextMenuItemButtonData data)
```

Parameters

TYPE	NAME	DESCRIPTION
ContextMenuItemButtonData	data	This is a ContextMenuitemData that will be used to create the menu items.

Interaction()

This will setup the Interaction when the mouse enters and leave the hover of this element.

Declaration

```
public override void Interaction()
```

Overrides

[ContextMenuElement.Interaction\(\)](#)

Style()

This will the style of the context menu.

Declaration

```
public override void Style()
```

Overrides

Extension Methods

VisualElementStyleExtension.BackgroundColor(VisualElement, Color)
VisualElementStyleExtension.BackgroundImage(VisualElement, Background)
VisualElementStyleExtension.BackgroundImage(VisualElement, Texture2D)
VisualElementStyleExtension.BackgroundImage(VisualElement, Sprite)
VisualElementStyleExtension.BackgroundImage(VisualElement, StyleKeyword)
VisualElementStyleExtension.Color(VisualElement, Color)
VisualElementStyleExtension.Color(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Top(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Right(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Left(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color(VisualElement, Color[])
VisualElementStyleExtension.Border_Color(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetRadius_Top(VisualElement)
VisualElementStyleExtension.GetRadius_Bottom(VisualElement)
VisualElementStyleExtension.GetRadius_Size(VisualElement)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius(VisualElement, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetBorder_Width(VisualElement)
VisualElementStyleExtension.GetBorder_Height(VisualElement)
VisualElementStyleExtension.GetBorder_Size(VisualElement)
VisualElementStyleExtension.Border_Width_Top(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Right(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Left(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width(VisualElement, Single[])
VisualElementStyleExtension.Border_Width(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Border(VisualElement, Single[])
VisualElementStyleExtension.Border(VisualElement, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])

VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)
VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)

VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)
VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)

VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)
VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)
VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)
VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)
VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)
VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)
VisualElementExtension.SetTransform_Position(VisualElement, Vector2)

VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)
VisualElementExtension.GetElementSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)
VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)
CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)
CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32, Boolean)
CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)
CreateElements.CreateElement_Icon(VisualElement, InventoryItem)

CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)

CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class ContextMenuItemButtonElement.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Inheritance

System.Object

UnityEngine.UIElements.UxmlFactory<[ContextMenuItemButtonElement](#), [VisualElement.UxmlTraits](#)>

ContextMenuItemButtonElement.UxmlFactory

Namespace: [UtilitySystem.UI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<ContextMenuItemButtonElement, VisualElement.UxmlTraits>, IUxmlFactory
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)

Class ContextMenuItemElement

This class is the context menu item for the runtime.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
ContextMenuItemElement
[ContextMenuItemButtonElement](#)
[ContextMenuItemParentElement](#)

Namespace: [UtilitySystem.UI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class ContextMenuItemElement : VisualElement, IEventHandler, ITransform, ITransitionAnimations, IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

ContextMenuItemElement()

This is a constructor and will create the context menu item.

Declaration

```
public ContextMenuItemElement()
```

ContextMenuItemElement(ContextMenuItemData)

This is a constructor and will create the context menu item.

Declaration

```
public ContextMenuItemElement(ContextMenuItemData data)
```

Parameters

TYPE	NAME	DESCRIPTION
ContextMenuItemData	data	this is the list of Context Menu Item Data

Fields

text

This is the label that will display the text of the item.

Declaration

```
public Label text
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Label	

Methods

GenerateElements(ContextMenuItemData)

This will generate "ContextMenuitem" elements for this menu

Declaration

```
public void GenerateElements(ContextMenuItemData data)
```

Parameters

TYPE	NAME	DESCRIPTION
ContextMenuItemData	data	This is a ContextMenuItemData that will be used to create the menu items.

Interaction()

This will setup the Interaction when the mouse enters and leave the hover of this element.

Declaration

```
public virtual void Interaction()
```

Style()

This will the style of the context menu.

Declaration

```
public virtual void Style()
```

Extension Methods

[VisualElementStyleExtension.BackgroundColor\(VisualElement, Color\)](#)

VisualElementStyleExtension.Background_Image(VisualElement, Background)
VisualElementStyleExtension.Background_Image(VisualElement, Texture2D)
VisualElementStyleExtension.Background_Image(VisualElement, Sprite)
VisualElementStyleExtension.Background_Image(VisualElement, StyleKeyword)
VisualElementStyleExtension.Color(VisualElement, Color)
VisualElementStyleExtension.Color(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Top(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Right(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Left(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color(VisualElement, Color[])
VisualElementStyleExtension.Border_Color(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetRadius_Top(VisualElement)
VisualElementStyleExtension.GetRadius_Bottom(VisualElement)
VisualElementStyleExtension.GetRadius_Size(VisualElement)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius(VisualElement, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetBorder_Width(VisualElement)
VisualElementStyleExtension.GetBorder_Height(VisualElement)
VisualElementStyleExtension.GetBorder_Size(VisualElement)
VisualElementStyleExtension.Border_Width_Top(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Right(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Left(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width(VisualElement, Single[])
VisualElementStyleExtension.Border_Width(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Border(VisualElement, Single[])
VisualElementStyleExtension.Border(VisualElement, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)

VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])

VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)
VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)
VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)

VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)
VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)
VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)
VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)
VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)
VisualElementExtension.SetTransform_Position(VisualElement, Vector2)
VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)

VisualElementExtension.GetSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)
VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)
CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)
CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32, Boolean)
CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)
CreateElements.CreateElement_Icon(VisualElement, InventoryItem)
CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)
CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class ContextMenuItemElement.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Inheritance

System.Object

UnityEngine.UIElements.UxmlFactory<[ContextMenuItemElement](#), [VisualElement.UxmlTraits](#)>

ContextMenuItemElement.UxmlFactory

Namespace: [UtilitySystem.UI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<ContextMenuItemElement, VisualElement.UxmlTraits>, IUxmlFactory
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)

Class ContextMenuItemParentElement

This class is the context sub-menu for the runtime.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
[ContextMenuItemElement](#)
ContextMenuItemParentElement

Namespace: [UtilitySystem.UI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class ContextMenuItemParentElement : ContextMenuItemElement, IEventHandler, ITransform, ITransitionAnimations, IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

ContextMenuItemParentElement()

This is a constructor and will create the context menu item.

Declaration

```
public ContextMenuItemParentElement()
```

ContextMenuItemParentElement(ContextMenuItemParentData)

This is a constructor and will create the context menu item.

Declaration

```
public ContextMenuItemParentElement(ContextMenuItemParentData data)
```

Parameters

TYPE	NAME	DESCRIPTION
ContextMenuItemParentData	data	this is the list of Context Menu Item Parent Data

Fields

arrow

This is the element that contains '>' that rotates when hovered.

Declaration

```
public VisualElement arrow
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.VisualElement	

menu

This is the sub menu attached to this item.

Declaration

```
public ContextMenuElement menu
```

Field Value

TYPE	DESCRIPTION
ContextMenuElement	

Methods

GenerateElements(ContextMenuItemParentData)

This will generate "ContextMenuItem" elements for this menu

Declaration

```
public void GenerateElements(ContextMenuItemParentData data)
```

Parameters

TYPE	NAME	DESCRIPTION
ContextMenuItemParentData	data	This is a ContextMenuItemData that will be used to create the menu items.

Style()

This will the style of the context menu.

Declaration

```
public override void Style()
```

Overrides

[ContextMenuElement.Style\(\)](#)

Extension Methods

[VisualElementStyleExtension.BackgroundColor\(VisualElement, Color\)](#)
[VisualElementStyleExtension.Background_Image\(VisualElement, Background\)](#)
[VisualElementStyleExtension.Background_Image\(VisualElement, Texture2D\)](#)
[VisualElementStyleExtension.Background_Image\(VisualElement, Sprite\)](#)
[VisualElementStyleExtension.Background_Image\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Color\(VisualElement, Color\)](#)
[VisualElementStyleExtension.Color\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Color_Top\(VisualElement, Color\)](#)
[VisualElementStyleExtension.Border_Color_Top\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Color_Right\(VisualElement, Color\)](#)
[VisualElementStyleExtension.Border_Color_Right\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Color_Bottom\(VisualElement, Color\)](#)
[VisualElementStyleExtension.Border_Color_Bottom\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Color_Left\(VisualElement, Color\)](#)
[VisualElementStyleExtension.Border_Color_Left\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Color\(VisualElement, Color\[\]\)](#)
[VisualElementStyleExtension.Border_Color\(VisualElement, StyleKeyword\[\]\)](#)
[VisualElementStyleExtension.GetRadius_Top\(VisualElement\)](#)
[VisualElementStyleExtension.GetRadius_Bottom\(VisualElement\)](#)
[VisualElementStyleExtension.GetRadius_Size\(VisualElement\)](#)
[VisualElementStyleExtension.Border_Radius_Top_Left\(VisualElement, Single, LengthUnit\)](#)
[VisualElementStyleExtension.Border_Radius_Top_Left\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Radius_Top_Right\(VisualElement, Single, LengthUnit\)](#)
[VisualElementStyleExtension.Border_Radius_Top_Right\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Radius_Bottom_Left\(VisualElement, Single, LengthUnit\)](#)
[VisualElementStyleExtension.Border_Radius_Bottom_Left\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Radius_Bottom_Right\(VisualElement, Single, LengthUnit\)](#)
[VisualElementStyleExtension.Border_Radius_Bottom_Right\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Radius\(VisualElement, Single\[\]\)](#)
[VisualElementStyleExtension.Border_Radius\(VisualElement, LengthUnit, Single\[\]\)](#)
[VisualElementStyleExtension.Border_Radius\(VisualElement, LengthUnit\[\], Single\[\]\)](#)
[VisualElementStyleExtension.Border_Radius\(VisualElement, StyleKeyword\[\]\)](#)
[VisualElementStyleExtension.GetBorder_Width\(VisualElement\)](#)
[VisualElementStyleExtension.GetBorder_Height\(VisualElement\)](#)
[VisualElementStyleExtension.GetBorder_Size\(VisualElement\)](#)
[VisualElementStyleExtension.Border_Width_Top\(VisualElement, Single\)](#)
[VisualElementStyleExtension.Border_Width_Top\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Width_Right\(VisualElement, Single\)](#)
[VisualElementStyleExtension.Border_Width_Right\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Width_Bottom\(VisualElement, Single\)](#)
[VisualElementStyleExtension.Border_Width_Bottom\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Width_Left\(VisualElement, Single\)](#)
[VisualElementStyleExtension.Border_Width_Left\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Width\(VisualElement, Single\[\]\)](#)
[VisualElementStyleExtension.Border_Width\(VisualElement, StyleKeyword\[\]\)](#)
[VisualElementStyleExtension.Border\(VisualElement, Single\[\]\)](#)
[VisualElementStyleExtension.Border\(VisualElement, Color, Single\[\]\)](#)
[VisualElementStyleExtension.Border\(VisualElement, Color\[\], Single\[\]\)](#)

VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)
VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)

VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)

VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)
VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)
VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)
VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)
VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)
VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)

VisualElementExtension.SetTransform_Position(VisualElement, Vector2)
VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)
VisualElementExtension.GetElementSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)
VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)
CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)
CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32, Boolean)
CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)

CreateElements.CreateElement_Icon(VisualElement, InventoryItem)

CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)

CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class ContextMenuItemParentElement.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Inheritance

System.Object

UnityEngine.UIElements.UxmlFactory<[ContextMenuItemParentElement](#), [VisualElement.UxmlTraits](#)>

ContextMenuItemParentElement.UxmlFactory

Namespace: [UtilitySystem.UI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<ContextMenuItemParentElement, VisualElement.UxmlTraits>, IUxmlFactory
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)

Class DialogBoxElement

This class is the dialog box that can be move around on the screen.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
DialogBoxElement

Namespace: [UtilitySystem.UI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class DialogBoxElement : VisualElement, IEventHandler, ITransform, ITransitionAnimations, IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

DialogBoxElement()

This is a constructor and will create the dialog box.

Declaration

```
public DialogBoxElement()
```

DialogBoxElement(String, String, Boolean, StyleSheet)

This is a constructor and will create the dialog box.

Declaration

```
public DialogBoxElement(string UID, string titleText, bool isDisplay = false, StyleSheet style = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	UID	This is the Unique ID for this DialogBox.
System.String	titleText	The text for the label that will be the title of the element.
System.Boolean	isDisplay	This boolean if true will add Visual to the onClose or Close if false.

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.StyleSheet	style	The stylesheet for the element.

DialogBoxElement(String, String, VisualElement, Boolean, StyleSheet)

This is a constructor and will create the dialog box.

Declaration

```
public DialogBoxElement(string UID, string titleText, VisualElement content, bool isDisplay = false,
StyleSheet style = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	UID	This is the Unique ID for this DialogBox.
System.String	titleText	The text for the label that will be the title of the element.
UnityEngine.UIElements.VisualElement	content	This element will be added to the content box.
System.Boolean	isDisplay	This boolean if true will add Visual to the onClose or Close if false.
UnityEngine.UIElements.StyleSheet	style	The stylesheet for the element.

Properties

close

This is the button that will be on the title bar.

Declaration

```
public Button close { get; }
```

Property Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

GetContentBox

This will get the contentBox.

Declaration

```
public VisualElement GetContentBox { get; }
```

Property Value

TYPE	DESCRIPTION
UnityEngine.UIElements.VisualElement	returns the contentBox.

GetFootBar

This will get the footBar.

Declaration

```
public VisualElement GetFootBar { get; }
```

Property Value

TYPE	DESCRIPTION
UnityEngine.UIElements.VisualElement	returns the footBar.

GetTitleBar

This will get the titlebar.

Declaration

```
public VisualElement GetTitleBar { get; }
```

Property Value

TYPE	DESCRIPTION
UnityEngine.UIElements.VisualElement	returns the titleBar.

maximise

This is the button that will be on the title bar.

Declaration

```
public Button maximise { get; }
```

Property Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

minimise

This is the button that will be on the title bar.

Declaration

```
public Button minimise { get; }
```

Property Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

UID

This is the unique id for this DialogBox

Declaration

```
public string UID { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

Close()

This will remove this element from it's parent.

Declaration

```
public void Close()
```

CloseAction(Action, Boolean)

This will add or remove an action for onClose

Declaration

```
public void CloseAction(Action action, bool isAdd = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action	action	This is the action that will be added or removed.
System.Boolean	isAdd	If this is true then add else remove.

Content(VisualElement, Boolean)

This will add, remove an element or clear the whole content box.

Declaration

```
public void Content(VisualElement content = null, bool isAdd = true)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	content	Is an optional element to be added to the content box.
System.Boolean	isAdd	Is an optional bool to determine of added or removed.

Footer(VisualElement, Boolean)

This will add, remove an element or clear the whole foot bar.

Declaration

```
public void Footer(VisualElement content = null, bool isAdd = true)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	content	Is an optional element to be added to the foot bar.
System.Boolean	isAdd	Is an optional bool to determine of added or removed.

GenerateButton(Action, String, String)

This will create the buttons for the titlebar.

Declaration

```
public Button GenerateButton(Action action, string name, string text)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action	action	This is the action that the button will use.
System.String	name	
System.String	text	

Returns

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	The Return is the button that was created.

HideAction(Action, Boolean)

This will add or remove an action for onHide

Declaration

```
public void HideAction(Action action, bool isAdd = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action	action	This is the action that will be added or removed.

TYPE	NAME	DESCRIPTION
System.Boolean	isAdd	If this is true then add else remove.

MaximiseAction(Action, Boolean)

This will add or remove an action for onMaximise

Declaration

```
public void MaximiseAction(Action action, bool isAdd = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action	action	This is the action that will be added or removed.
System.Boolean	isAdd	If this is true then add else remove.

MinimiseAction(Action, Boolean)

This will add or remove an action for onMinimise

Declaration

```
public void MinimiseAction(Action action, bool isAdd = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action	action	This is the action that will be added or removed.
System.Boolean	isAdd	If this is true then add else remove.

ShowAction(Action, Boolean)

This will add or remove an action for onShow

Declaration

```
public void ShowAction(Action action, bool isAdd = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action	action	This is the action that will be added or removed.
System.Boolean	isAdd	If this is true then add else remove.

Title(VisualElement, Boolean)

This will add, remove an element or clear the whole title bar.

Declaration

```
public void Title(VisualElement content = null, bool isAdd = true)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.UIElements.VisualElement	content	Is an optional element to be added to the title bar.
System.Boolean	isAdd	Is an optional bool to determine if added or removed.

Visual(Boolean)

This allows for the changing of the display of the dialogbox so it is kept instead of removing the UI and can be turned on if needed.

Declaration

```
public void Visual(bool show = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	show	This boolean if true will invoke the onShow() action or onHide() if false.

WindowDragInit(Action<Object>)

This when invoked will allow for the element to be moved by the player.

Declaration

```
public void WindowDragInit(Action<object> action = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action<System.Object>	action	

Remarks

Parent Element needed and will cover the whole screen on mouse down event.

Extension Methods

- VisualElementStyleExtension.BackGround_Color(VisualElement, Color)
- VisualElementStyleExtension.Background_Image(VisualElement, Background)
- VisualElementStyleExtension.Background_Image(VisualElement, Texture2D)
- VisualElementStyleExtension.Background_Image(VisualElement, Sprite)
- VisualElementStyleExtension.Background_Image(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Color(VisualElement, Color)
- VisualElementStyleExtension.Color(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Color_Top(VisualElement, Color)
- VisualElementStyleExtension.Border_Color_Top(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Color_Right(VisualElement, Color)
- VisualElementStyleExtension.Border_Color_Right(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Color_Bottom(VisualElement, Color)
- VisualElementStyleExtension.Border_Color_Bottom(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Color_Left(VisualElement, Color)
- VisualElementStyleExtension.Border_Color_Left(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Color(VisualElement, Color[])
- VisualElementStyleExtension.Border_Color(VisualElement, StyleKeyword[])
- VisualElementStyleExtension.GetRadius_Top(VisualElement)
- VisualElementStyleExtension.GetRadius_Bottom(VisualElement)
- VisualElementStyleExtension.GetRadius_Size(VisualElement)
- VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, Single, LengthUnit)
- VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, Single, LengthUnit)
- VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, Single, LengthUnit)
- VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)
- VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Radius(VisualElement, Single[])
- VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit, Single[])
- VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit[], Single[])
- VisualElementStyleExtension.Border_Radius(VisualElement, StyleKeyword[])
- VisualElementStyleExtension.GetBorder_Width(VisualElement)
- VisualElementStyleExtension.GetBorder_Height(VisualElement)
- VisualElementStyleExtension.GetBorder_Size(VisualElement)
- VisualElementStyleExtension.Border_Width_Top(VisualElement, Single)
- VisualElementStyleExtension.Border_Width_Top(VisualElement, StyleKeyword)

VisualElementStyleExtension.Border_Width_Right(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Left(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width(VisualElement, Single[])
VisualElementStyleExtension.Border_Width(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Border(VisualElement, Single[])
VisualElementStyleExtension.Border(VisualElement, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)
VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])

VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)

VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)
VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)
VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)
VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)
VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)
VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)

VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)
VisualElementExtension.SetTransform_Position(VisualElement, Vector2)
VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)
VisualElementExtension.GetElementSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)
VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)
CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)
CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)

CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32, Boolean)
CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)
CreateElements.CreateElement_Icon(VisualElement, InventoryItem)
CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)
CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class DialogBoxElement.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Inheritance

System.Object

UnityEngine.UIElements.UxmlFactory<[DialogBoxElement](#), [VisualElement.UxmlTraits](#)>

DialogBoxElement.UxmlFactory

Namespace: [UtilitySystem.UI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<DialogBoxElement, VisualElement.UxmlTraits>, IUxmlFactory
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)

Class PopupBaseElement

This abstract class that is the base of the popup elements.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
PopupBaseElement
[SlotPopupElement](#)
[TransferPopupElement](#)

Namespace: [UtilitySystem.UI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public abstract class PopupBaseElement : VisualElement, IEventHandler, ITransform, ITransitionAnimations, IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

PopupBaseElement()

This is the constructor

Declaration

```
public PopupBaseElement()
```

PopupBaseElement(Vector2)

This is the constructor

Declaration

```
public PopupBaseElement(Vector2 size)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	size	This is the size of the popup.

Fields

background

This is the background of the popup.

Declaration

```
protected VisualElement background
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.VisualElement	

buttonParent

This is the element that holds the button.

Declaration

```
protected VisualElement buttonParent
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.VisualElement	

cancelButton

This is the cancel button.

Declaration

```
protected Button cancelButton
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

contentParent

This is the element that will hold the content.

Declaration

```
protected VisualElement contentParent
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.VisualElement	

submitButton

This is the submit button.

Declaration

```
protected Button submitButton
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Button	

titleText

This is the title of the popup.

Declaration

```
protected Label titleText
```

Field Value

TYPE	DESCRIPTION
UnityEngine.UIElements.Label	

Methods

Cancel()

This will be invoke by the cancel button and will close the popup.

Declaration

```
public virtual void Cancel()
```

GenerateElements()

This will generate the elements for the popup

Declaration

```
public virtual void GenerateElements()
```

Style(Vector2)

This will style the element.

Declaration

```
public virtual void Style(Vector2 size)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	size	

Submit()

This will be invoke by the submit button and will either drop or split the item.

Declaration

```
public abstract void Submit()
```

Extension Methods

[VisualElementStyleExtension.BackgroundColor\(VisualElement, Color\)](#)
[VisualElementStyleExtension.BackgroundImage\(VisualElement, Background\)](#)
[VisualElementStyleExtension.BackgroundImage\(VisualElement, Texture2D\)](#)
[VisualElementStyleExtension.BackgroundImage\(VisualElement, Sprite\)](#)
[VisualElementStyleExtension.BackgroundImage\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Color\(VisualElement, Color\)](#)
[VisualElementStyleExtension.Color\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Color_Top\(VisualElement, Color\)](#)
[VisualElementStyleExtension.Border_Color_Top\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Color_Right\(VisualElement, Color\)](#)
[VisualElementStyleExtension.Border_Color_Right\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Color_Bottom\(VisualElement, Color\)](#)
[VisualElementStyleExtension.Border_Color_Bottom\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Color_Left\(VisualElement, Color\)](#)
[VisualElementStyleExtension.Border_Color_Left\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Color\(VisualElement, Color\[\]\)](#)
[VisualElementStyleExtension.Border_Color\(VisualElement, StyleKeyword\[\]\)](#)
[VisualElementStyleExtension.GetRadius_Top\(VisualElement\)](#)
[VisualElementStyleExtension.GetRadius_Bottom\(VisualElement\)](#)
[VisualElementStyleExtension.GetRadius_Size\(VisualElement\)](#)
[VisualElementStyleExtension.Border_Radius_Top_Left\(VisualElement, Single, LengthUnit\)](#)
[VisualElementStyleExtension.Border_Radius_Top_Left\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Radius_Top_Right\(VisualElement, Single, LengthUnit\)](#)
[VisualElementStyleExtension.Border_Radius_Top_Right\(VisualElement, StyleKeyword\)](#)
[VisualElementStyleExtension.Border_Radius_Bottom_Left\(VisualElement, Single, LengthUnit\)](#)

VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius(VisualElement, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetBorder_Width(VisualElement)
VisualElementStyleExtension.GetBorder_Height(VisualElement)
VisualElementStyleExtension.GetBorder_Size(VisualElement)
VisualElementStyleExtension.Border_Width_Top(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Right(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Left(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width(VisualElement, Single[])
VisualElementStyleExtension.Border_Width(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Border(VisualElement, Single[])
VisualElementStyleExtension.Border(VisualElement, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)
VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)

VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)

VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)
VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)
VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)
VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)

VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)
VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)
VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)
VisualElementExtension.SetTransform_Position(VisualElement, Vector2)
VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)
VisualElementExtension.GetElementSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)
VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)

CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)
CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32, Boolean)
CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)
CreateElements.CreateElement_Icon(VisualElement, InventoryItem)
CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)
CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class SpriteElement

This Class is an VisualElement that is created for the Inventory Editor.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
UnityEngine.UIElements.Image
SpriteElement

Namespace: [UtilitySystem.UI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class SpriteElement : Image, IEventHandler, ITransform, ITransitionAnimations, IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

SpriteElement()

This is a constructor and will set the style for the SpriteElement.

Declaration

```
public SpriteElement()
```

SpriteElement(Sprite, StyleSheet)

This is a constructor and will set the style for the SpriteElement.

Declaration

```
public SpriteElement(Sprite sprite, StyleSheet style = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Sprite	sprite	
UnityEngine.UIElements.StyleSheet	style	

Methods

RemoveSprite()

This will remove the sprite for this element.

Declaration

```
public void RemoveSprite()
```

SetSprite(Sprite)

This will set the sprite for this element.

Declaration

```
public void SetSprite(Sprite sprite)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Sprite	sprite	

Extension Methods

- VisualElementStyleExtension.BackgroundColor(VisualElement, Color)
- VisualElementStyleExtension.BackgroundImage(VisualElement, Background)
- VisualElementStyleExtension.BackgroundImage(VisualElement, Texture2D)
- VisualElementStyleExtension.BackgroundImage(VisualElement, Sprite)
- VisualElementStyleExtension.BackgroundImage(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Color(VisualElement, Color)
- VisualElementStyleExtension.Color(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Color_Top(VisualElement, Color)
- VisualElementStyleExtension.Border_Color_Top(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Color_Right(VisualElement, Color)
- VisualElementStyleExtension.Border_Color_Right(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Color_Bottom(VisualElement, Color)
- VisualElementStyleExtension.Border_Color_Bottom(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Color_Left(VisualElement, Color)
- VisualElementStyleExtension.Border_Color_Left(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Color(VisualElement, Color[])
- VisualElementStyleExtension.Border_Color(VisualElement, StyleKeyword[])
- VisualElementStyleExtension.GetRadius_Top(VisualElement)
- VisualElementStyleExtension.GetRadius_Bottom(VisualElement)
- VisualElementStyleExtension.GetRadius_Size(VisualElement)
- VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, Single, LengthUnit)
- VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, Single, LengthUnit)
- VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, Single, LengthUnit)
- VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)
- VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, StyleKeyword)
- VisualElementStyleExtension.Border_Radius(VisualElement, Single[])
- VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit, Single[])

VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetBorder_Width(VisualElement)
VisualElementStyleExtension.GetBorder_Height(VisualElement)
VisualElementStyleExtension.GetBorder_Size(VisualElement)
VisualElementStyleExtension.Border_Width_Top(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Right(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Left(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width(VisualElement, Single[])
VisualElementStyleExtension.Border_Width(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Border(VisualElement, Single[])
VisualElementStyleExtension.Border(VisualElement, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)
VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)

VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)

VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)
VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)
VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)
VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)
VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)

VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)
VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)
VisualElementExtension.SetTransform_Position(VisualElement, Vector2)
VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)
VisualElementExtension.GetElementSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)
VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)
CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)

CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32, Boolean)
CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)
CreateElements.CreateElement_Icon(VisualElement, InventoryItem)
CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)
CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class SpriteElement.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Inheritance

System.Object

UnityEngine.UIElements.UxmlFactory<[SpriteElement](#), [Image.UxmlTraits](#)>

SpriteElement.UxmlFactory

Namespace: [UtilitySystem.UI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<SpriteElement, Image.UxmlTraits>, IUxmlFactory
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)

Class ToolTipElement

This class will create a tooltip for your game.

Inheritance

System.Object
UnityEngine.UIElements.CallbackEventHandler
UnityEngine.UIElements.Focusable
UnityEngine.UIElements.VisualElement
ToolTipElement

Namespace: [UtilitySystem.UI](#)
Assembly: Assembly-CSharp.dll

Syntax

```
public class ToolTipElement : VisualElement, IEventHandler, ITransform, ITransitionAnimations, IExperimentalFeatures, IVisualElementScheduler, IResolvedStyle
```

Constructors

ToolTipElement()

This is a constructor for the tool tip.

Declaration

```
public ToolTipElement()
```

ToolTipElement(Font, Int32, StyleSheet)

This is a constructor for the tool tip.

Declaration

```
public ToolTipElement(Font font = null, int textSize = 0, StyleSheet style = null)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Font	font	Is an optional font that will be added to the label.
System.Int32	textSize	Is an optional size of the text in the label.
UnityEngine.UIElements.StyleSheet	style	Is an optional stylesheet for the element.

Methods

AddText(List<Func<String>>)

This will add a new message to the Func string list.

Declaration

```
public void AddText(List<Func<string>> messages)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.List<System.Func<System.String>>	messages	The list of func strings that will be added to the func string list for the message.

AddText(Func<String>)

This will add a new message to the Func string list.

Declaration

```
public void AddText(Func<string> message)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Func<System.String>	message	The func string that will be added to the func string list for the message.

AddText(String)

This will add a new message to the Func string list.

Declaration

```
public void AddText(string message)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	The string that will be added to the func string list for the message.

RemoveText()

this will clear the message.

Declaration

```
public void RemoveText()
```

Style()

This will set the style of the Icon.

Declaration

```
public void Style()
```

TextSize(Int32)

This will allow for the text size to be changed.

Declaration

```
public void TextSize(int textSize)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	textSize	the int used to update the text size.

UpdatePosition(Vector2)

This will update the position of the element on screen and maintain its bound within the screens bounds.

Declaration

```
public void UpdatePosition(Vector2 mousePosition)
```

Parameters

TYPE	NAME	DESCRIPTION
UnityEngine.Vector2	mousePosition	This is the vector2 position it is trying to move to.

UpdateText()

This will update the message that is displayed.

Declaration

```
public void UpdateText()
```

Extension Methods

```
VisualElementStyleExtension.BackgroundColor(VisualElement, Color)
VisualElementStyleExtension.BackgroundImage(VisualElement, Background)
VisualElementStyleExtension.BackgroundImage(VisualElement, Texture2D)
VisualElementStyleExtension.BackgroundImage(VisualElement, Sprite)
VisualElementStyleExtension.BackgroundImage(VisualElement, StyleKeyword)
VisualElementStyleExtension.Color(VisualElement, Color)
VisualElementStyleExtension.Color(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Top(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Right(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color_Left(VisualElement, Color)
VisualElementStyleExtension.Border_Color_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Color(VisualElement, Color[])
VisualElementStyleExtension.Border_Color(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetRadius_Top(VisualElement)
VisualElementStyleExtension.GetRadius_Bottom(VisualElement)
VisualElementStyleExtension.GetRadius_Size(VisualElement)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Top_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Border_Radius_Bottom_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Radius(VisualElement, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Border_Radius(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetBorder_Width(VisualElement)
VisualElementStyleExtension.GetBorder_Height(VisualElement)
VisualElementStyleExtension.GetBorder_Size(VisualElement)
VisualElementStyleExtension.Border_Width_Top(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Right(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width_Left(VisualElement, Single)
VisualElementStyleExtension.Border_Width_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Border_Width(VisualElement, Single[])
VisualElementStyleExtension.Border_Width(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Border(VisualElement, Single[])
VisualElementStyleExtension.Border(VisualElement, Color, Single[])
```

VisualElementStyleExtension.Border(VisualElement, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit, Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color, Single[])
VisualElementStyleExtension.Border(VisualElement, LengthUnit[], Color[], Single[])
VisualElementStyleExtension.Border(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetMargin_Width(VisualElement)
VisualElementStyleExtension.GetMargin_Height(VisualElement)
VisualElementStyleExtension.GetMargin_Size(VisualElement)
VisualElementStyleExtension.Margin_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Margin_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Margin(VisualElement, Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Margin(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Margin(VisualElement, StyleKeyword[])
VisualElementStyleExtension.GetPadding_Width(VisualElement)
VisualElementStyleExtension.GetPadding_Height(VisualElement)
VisualElementStyleExtension.GetPadding_Size(VisualElement)
VisualElementStyleExtension.Padding_Top(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Right(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Bottom(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding_Left(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Padding_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Padding(VisualElement, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Padding(VisualElement, LengthUnit[], Single[])
VisualElementStyleExtension.Padding(VisualElement, StyleKeyword[])
VisualElementStyleExtension.Size_Width(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size(VisualElement, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Min(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Min(VisualElement, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Min(VisualElement, Vector2, LengthUnit[])

VisualElementStyleExtension.Size_Min(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Min(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.Size_Width_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Width_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Height_Max(VisualElement, Single, LengthUnit)
VisualElementStyleExtension.Size_Height_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.Size_Max(VisualElement, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, LengthUnit, LengthUnit, Single[])
VisualElementStyleExtension.Size_Max(VisualElement, Vector2, LengthUnit[])
VisualElementStyleExtension.Size_Max(VisualElement, StyleKeyword)
VisualElementStyleExtension.GetSize_Max(VisualElement, Boolean, Boolean)
VisualElementStyleExtension.SetDisplay(VisualElement, DisplayStyle)
VisualElementStyleExtension.GetDisplay(VisualElement)
VisualElementStyleExtension.Display_ToggleVisibility(VisualElement)
VisualElementStyleExtension.Display_Hide(VisualElement)
VisualElementStyleExtension.Display_Show(VisualElement)
VisualElementStyleExtension.Display_IsInvisible(VisualElement)
VisualElementStyleExtension.Display_IsVisible(VisualElement)
VisualElementStyleExtension.StylePosition(VisualElement, Position)
VisualElementStyleExtension.StylePosition(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Toggle(VisualElement)
VisualElementStyleExtension.StylePosition_Top(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Top(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Right(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Right(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Bottom(VisualElement, StyleKeyword)
VisualElementStyleExtension.StylePosition_Left(VisualElement, Single)
VisualElementStyleExtension.StylePosition_Left(VisualElement, StyleKeyword)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Top_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Left(VisualElement, Single, Single)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Vector2)
VisualElementStyleExtension.Position_Bottom_Right(VisualElement, Single, Single)
VisualElementStyleExtension.Anchored_Position_Top_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Top_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Left(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Anchored_Position_Bottom_Right(VisualElement, UIAnchorEnum)
VisualElementStyleExtension.Align_Content(VisualElement, Align)
VisualElementStyleExtension.Align_Content(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Item(VisualElement, Align)
VisualElementStyleExtension.Align_Item(VisualElement, StyleKeyword)
VisualElementStyleExtension.Align_Self(VisualElement, Align)
VisualElementStyleExtension.Align_Self(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Wrap(VisualElement, Wrap)
VisualElementStyleExtension.Flex_Wrap(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Shrink(VisualElement, Single)
VisualElementStyleExtension.Flex_Shrink(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Grow(VisualElement, Single)
VisualElementStyleExtension.Flex_Grow(VisualElement, StyleKeyword)

VisualElementStyleExtension.Flex_Direction(VisualElement, FlexDirection)
VisualElementStyleExtension.Flex_Direction(VisualElement, StyleKeyword)
VisualElementStyleExtension.Flex_Basis(VisualElement, Single)
VisualElementStyleExtension.Flex_Basis(VisualElement, StyleKeyword)
VisualElementStyleExtension.White_Space(VisualElement, WhiteSpace)
VisualElementStyleExtension.White_Space(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font_Size(VisualElement, Single)
VisualElementStyleExtension.Font_Size(VisualElement, StyleKeyword)
VisualElementStyleExtension.Font(VisualElement, Font)
VisualElementStyleExtension.Font(VisualElement, FontAsset)
VisualElementStyleExtension.Text_Position(VisualElement, TextAnchor)
VisualElementStyleExtension.Overflow(VisualElement, Overflow)
VisualElementStyleExtension.Overflow(VisualElement, StyleKeyword)
VisualElementExtension.ViewDataKey(VisualElement, String)
VisualElementExtension.UsageHints(VisualElement, UsageHints)
VisualElementExtension.ToolTip(VisualElement, String)
VisualElementExtension.GetLayout(VisualElement)
VisualElementExtension.GetLayoutSize(VisualElement)
VisualElementExtension.GetLayoutWidth(VisualElement)
VisualElementExtension.GetLayoutHeight(VisualElement)
VisualElementExtension.GetLayoutPosition(VisualElement)
VisualElementExtension.GetLayoutPositionX(VisualElement)
VisualElementExtension.GetLayoutPositionY(VisualElement)
VisualElementExtension.GetWorldBound(VisualElement)
VisualElementExtension.GetWorldBoundSize(VisualElement)
VisualElementExtension.GetWorldBoundPosition(VisualElement)
VisualElementExtension.GetLocalBound(VisualElement)
VisualElementExtension.GetLocalBoundSize(VisualElement)
VisualElementExtension.GetLocalBoundPosition(VisualElement)
VisualElementExtension.Pick(VisualElement, PickingMode)
VisualElementExtension.Pick_Toggle(VisualElement)
VisualElementExtension.Pick_OFF(VisualElement)
VisualElementExtension.Pick_ON(VisualElement)
VisualElementExtension.Focus(VisualElement, Boolean)
VisualElementExtension.Focus_Toggle(VisualElement)
VisualElementExtension.Focus_OFF(VisualElement)
VisualElementExtension.Focus_ON(VisualElement)
VisualElementExtension.SetVisible(VisualElement, Boolean)
VisualElementExtension.GetVisible(VisualElement)
VisualElementExtension.Visible_ToggleVisibility(VisualElement)
VisualElementExtension.Visible_Hide(VisualElement)
VisualElementExtension.Visible_Show(VisualElement)
VisualElementExtension.Visible_IsInvisible(VisualElement)
VisualElementExtension.Visible_IsVisible(VisualElement)
VisualElementExtension.ResetRotation(VisualElement)
VisualElementExtension.SetTransform_Rotate(VisualElement, Vector3)
VisualElementExtension.GetTransform_Rotate(VisualElement)
VisualElementExtension.Rotate_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Counter_Clockwise(VisualElement, Vector3)
VisualElementExtension.Rotate_Clockwise_X(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_X(VisualElement, Single)
VisualElementExtension.RotateY_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Counter_Clockwise_Y(VisualElement, Single)
VisualElementExtension.Rotate_Clockwise_Z(VisualElement, Single)

VisualElementExtension.Rotate_Counter_Clockwise_Z(VisualElement, Single)
VisualElementExtension.SetTransform_Position(VisualElement, Vector2)
VisualElementExtension.GetTransform_Position(VisualElement)
VisualElementExtension.SetTransform_Scale(VisualElement, Vector3)
VisualElementExtension.GetTransform_Scale(VisualElement)
VisualElementExtension.Child(VisualElement, VisualElement, Boolean)
VisualElementExtension.StyleSheet(VisualElement, StyleSheet, Boolean)
VisualElementExtension.Name(VisualElement, String)
VisualElementExtension.GetElementSize(VisualElement)
VisualElementExtension.ScreenBound(VisualElement)
VisualElementExtension.MouseGrab(VisualElement, Vector2)
VisualElementExtension.MouseRelease(VisualElement)
VisualElementExtension.UpdatePosition(VisualElement, Vector2)
RichText.UpperCaseText(Object)
RichText.LowerCaseText(Object)
RichText.BoldText(Object)
RichText.ItalicText(Object)
RichText.Text(Object)
RichText.UnderLineText(Object)
RichText.StrikeText(Object)
RichText.SupText(Object)
RichText.SubText(Object)
RichText.PositionText(Object, Single)
RichText.SizeText(Object, Int32)
RichText.ColoredText(Object, String)
RichText.ColoredText(Object, Color)
RichText.BooleanText(Object, Boolean)
RichText.BooleanText(Object, Boolean, Color)
CreateElements.CreateElement_Label(VisualElement, String, Boolean)
CreateElements.CreateElement_VisualElement(VisualElement)
CreateElements.CreateElement_ScrollView(VisualElement, ScrollViewMode, VisualElement[])
CreateElements.CreateElement_ListView<TValue>(VisualElement, List<TValue>, Single, SelectionType)
CreateElements.CreateElement_Button(VisualElement, Action)
CreateElements.CreateElement_Toggle(VisualElement, String)
CreateElements.CreateElement_TextField(VisualElement, String, Int32, Boolean, Boolean, Char)
CreateElements.CreateElement_Foldout(VisualElement, String, Boolean)
CreateElements.CreateElement_Slider(VisualElement, Single, Single, String)
CreateElements.CreateElement_Slider_Int(VisualElement, Int32, Int32, String)
CreateElements.CreateElement_Slider_Min_Max(VisualElement, String, Single, Single, Single, Single)
CreateElements.CreateElement_Progress_Bar(VisualElement, String, Single, Single, Single)
CreateElements.CreateElement_Dropdown(VisualElement, String, List<String>, Int32)
CreateElements.CreateElement_Radio(VisualElement, Boolean, String, Boolean)
CreateElements.CreateElement_Radio_Group(VisualElement, String, String[])
CreateElements.CreateElement_ContextMenuItem(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu(VisualElement, ContextMenuItemData[])
CreateElements.CreateElement_ContextMenu(VisualElement, List<ContextMenuItemData>)
CreateElements.CreateElement_ContextMenu_Item_Parent(VisualElement, ContextMenuItemParentData)
CreateElements.CreateElement_ContextMenu_Item(VisualElement, ContextMenuItemData)
CreateElements.CreateElement_ContextMenu_Item_Button(VisualElement, ContextMenuItemButtonData)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, Boolean)
CreateElements.CreateElement_DialogBox(VisualElement, String, String, VisualElement, Boolean)
CreateElements.CreateElement_Tooltip(VisualElement, Font, Int32)
CreateElements.CreateElement_Sprite(VisualElement, Sprite)
CreateElements.CreateElement_Slot(VisualElement, Int32, Boolean)

CreateElements.CreateElement_Inventory(VisualElement, Int32, Int32, Inventory)

CreateElements.CreateElement_Icon(VisualElement, InventoryItem)

CreateElements.CreateElement_SlotPopup(VisualElement, Vector2, Int32, InventoryElement, Boolean)

CreateElements.CreateElement_TransferPopup(VisualElement, Vector2, TransferItem, InventoryElement, InventoryElement)

Class TooltipElement.UxmlFactory

This new class is UxmlFactory and is needed to create a uxml tag for the UIDocument to read.

Inheritance

System.Object

UnityEngine.UIElements.UxmlFactory<[TooltipElement](#), [VisualElement.UxmlTraits](#)>

TooltipElement.UxmlFactory

Namespace: [UtilitySystem.UI](#)

Assembly: Assembly-CSharp.dll

Syntax

```
public class UxmlFactory : UxmlFactory<TooltipElement, VisualElement.UxmlTraits>, IUxmlFactory
```

Extension Methods

[RichText.UpperCaseText\(Object\)](#)

[RichText.LowerCaseText\(Object\)](#)

[RichText.BoldText\(Object\)](#)

[RichText.ItalicText\(Object\)](#)

[RichText.Text\(Object\)](#)

[RichText.UnderLineText\(Object\)](#)

[RichText.StrikeText\(Object\)](#)

[RichText.SupText\(Object\)](#)

[RichText.SubText\(Object\)](#)

[RichText.PositionText\(Object, Single\)](#)

[RichText.SizeText\(Object, Int32\)](#)

[RichText.ColoredText\(Object, String\)](#)

[RichText.ColoredText\(Object, Color\)](#)

[RichText.BooleanText\(Object, Boolean\)](#)

[RichText.BooleanText\(Object, Boolean, Color\)](#)