Unity Editor Utilities Reference

PiRho Soft

AngleDisplayDrawer

PiRhoSoft.UtilityEditor.AngleDisplayDrawer : PropertyDrawer

Description

Draws an editor for a Quaternion. Use the AngleDisplayAttribute to automatically use this drawer or the static methods to draw a Quaternion manually (i.e from a drawer, control, or editor).

Static Methods

float GetHeight(GUIContent *label*, **AngleDisplayType** *type*)

Returns the height the drawer needs to display a Quaternion with the specified label and type.

void Draw(GUIContent *label*, **SerializedProperty** *property*, **AngleDisplayType** *type*)

Draws the editor for a property referencing a Quaternion field using the specified label and type with automatic layout.

void Draw(Rect *position*, **GUIContent** *label*, **SerializedProperty** *property*, **AngleDisplayType** *type*)

Draws the editor for a property referencing a Quaternion field using the specified label and type at the specified position and size.

Quaternion Draw(GUIContent *label*, **Quaternion** *quaternion*, **AngleDisplayType** *type*)

Draws the editor for a Quaternion using the specified label and type with automatic layout. Changes are reflected in the return value.

Quaternion Draw(Rect *position*, **GUIContent** *label*, **Quaternion** *quaternion*, **AngleDisplayType** *type*)

Draws the editor for a Quaternion using the specified label and type at the specified position and size. Changes are reflected in the return value.

AssetDisplayDrawer

PiRhoSoft.UtilityEditor.AssetDisplayDrawer : PropertyDrawer

Description

Static Methods

float GetHeight(GUIContent *label***)**

void Draw(GUIContent *label***, SerializedProperty** *property***, Type** *assetType***, bool** *showNoneOption***, bool** *showEditButton***, AssetDisplaySaveLocation** *saveLocation***, string** *defaultName***)**

void Draw(Rect *position***, GUIContent** *label***, SerializedProperty** *property***, Type** *assetType,* **bool** *showNoneOption***, bool** *showEditButton***, AssetDisplaySaveLocation** *saveLocation***, string** *defaultName***)**

AssetType Draw<AssetType>(GUIContent *label***, AssetType** *asset***, bool** *showNoneOption***, bool** *showEditButton***, AssetDisplaySaveLocation** *saveLocation***, string** *defaultName***)**

AssetType Draw<AssetType>(Rect *position***, GUIContent** *label***, AssetType** *asset***, bool** *showNoneOption***, bool** *showEditButton***, AssetDisplaySaveLocation** *saveLocation***, string** *defaultName***)**

Object Draw(GUIContent *label***, Object** *asset***, Type** *assetType***, bool** *showNoneOption***, bool** *showEditButton***, AssetDisplaySaveLocation** *saveLocation***, string** *defaultName***)**

Object Draw(Rect *position***, GUIContent** *label***, Object** *asset***, Type** *assetType***, bool** *showNoneOption***, bool** *showEditButton***, AssetDisplaySaveLocation** *saveLocation***, string** *defaultName***)**

   # Constructors

AssetDisplayDrawer()

   # Public Methods

float GetPropertyHeight(SerializedProperty *property***, GUIContent** *label***)** *(virtual)*

void OnGUI(Rect *position***, SerializedProperty** *property***, GUIContent** *label***)** *(virtual)*

   <<<

AssetHelper

PiRhoSoft.UtilityEditor.AssetHelper : AssetPostprocessor

Description

Static Methods

AssetType CreateAsset\<AssetType>(string *name***)**

AssetType GetOrCreateAsset\<AssetType>(string *name***)**

ScriptableObject CreateAsset(string *name***, Type** *type***)**

ScriptableObject GetOrCreateAsset(string *name***, Type** *type***)**

ScriptableObject CreateAssetAtPath(string *path***, Type** *type***)**

AssetType GetAsset\<AssetType>()

AssetType GetAssetWithId\<AssetType>(string *id***)**

AssetType GetAssetAtPath\<AssetType>(string *path***)**

Object GetAsset(Type *assetType***)**

Object GetAssetWithId(string *id***, Type** *type***)**

Object GetAssetAtPath(string *path***, Type** *type***)**

List\<AssetType> ListAssets\<AssetType>()

IEnumerable\<AssetType> FindAssets\<AssetType>()

List\<Object> ListAssets(Type *assetType***)**

IEnumerable\<Object> FindAssets(Type *assetType***)**

AssetList GetAssetList\<AssetType>(bool *includeNone***, bool** *includeCreate***)**

AssetList GetAssetList(Type *assetType***, bool** *includeNone***, bool** *includeCreate***)**

string FindCommonPath(IEnumerable\<string> *paths***)**

   # Constructors

AssetHelper()

   <<<

AssetList

PiRhoSoft.UtilityEditor.AssetList

Description

Constructors

AssetList()

   # Public Fields

Type *Type*

bool *HasNone*

bool *HasCreate*

GUIContent[] *Names*

List\<Object> *Assets*

TypeList *Types*

   # Public Methods

int GetIndex(Object *asset***)**
Object GetAsset(int *index***)**
Type GetType(int *index***)**

    <<<

AssetNameControl

PiRhoSoft.UtilityEditor.AssetNameControl : PropertyScopeControl

Description

Constructors

AssetNameControl()

    # Public Methods

void Setup(SerializedProperty *property***, FieldInfo** *fieldInfo***, PropertyAttribute** *attribute***)**
(virtual)

float GetHeight(SerializedProperty *property***, GUIContent** *label***)** *(virtual)*

void Draw(Rect *position***, SerializedProperty** *property***, GUIContent** *label***)** *(virtual)*

    <<<

AssetNameDrawer

PiRhoSoft.UtilityEditor.AssetNameDrawer : ControlDrawer<AssetNameControl>

Description

Constructors

AssetNameDrawer()

    <<<

BoolPreference

PiRhoSoft.UtilityEditor.BoolPreference

Description

Constructors

BoolPreference(string *name***, bool** *defaultValue***)**

    # Public Properties

bool *Value*

 <<<

Button

PiRhoSoft.UtilityEditor.Button

Description

Constructors

Button(Icon *icon*, **string** *label*, **string** *tooltip***)**

 # Public Properties

GUIContent *Content (read only)*

 <<<

ClassDisplayDrawer

PiRhoSoft.UtilityEditor.ClassDisplayDrawer : PropertyDrawer

Description

Static Methods

float GetHeight(SerializedProperty *property***)**
void Draw(SerializedProperty *property*, **GUIContent** *label***)**
void Draw(Rect *position*, **SerializedProperty** *property*, **GUIContent** *label***)**

 # Constructors

ClassDisplayDrawer()

 # Public Methods

float GetPropertyHeight(SerializedProperty *property*, **GUIContent** *label***)** *(virtual)*
void OnGUI(Rect *position*, **SerializedProperty** *property*, **GUIContent** *label***)** *(virtual)*

 <<<

ColorScope

PiRhoSoft.UtilityEditor.ColorScope : Scope

Description

Static Methods

ColorScope Color(Color *color***)**

ColorScope BackgroundColor(Color *backgroundColor***)**

ColorScope ContentColor(Color *contentColor***)**

   # Constructors

ColorScope(Color *color***, Color** *backgroundColor***, Color** *contentColor***)**

   # Protected Methods

void CloseScope() *(virtual)*

   <<<

ConditionalDisplayControl

PiRhoSoft.UtilityEditor.ConditionalDisplayControl : PropertyScopeControl

Description

Public Methods

void Setup(SerializedProperty *property***, FieldInfo** *fieldInfo***, PropertyAttribute** *attribute***)**
(virtual)

   # Protected Constructors

ConditionalDisplayControl()

   # Protected Fields

ConditionalDisplayAttribute *_attribute*

   # Protected Methods

bool IsVisible(SerializedProperty *property***)**

   <<<

ConditionalDisplayOtherControl

PiRhoSoft.UtilityEditor.ConditionalDisplayOtherControl : ConditionalDisplayControl

Description

Constructors

ConditionalDisplayOtherControl()

   # Public Methods

float GetHeight(SerializedProperty *property*, **GUIContent** *label***)** *(virtual)*

void Draw(Rect *position*, **SerializedProperty** *property*, **GUIContent** *label***)** *(virtual)*

   <<<

ConditionalDisplayOtherDrawer

PiRhoSoft.UtilityEditor.ConditionalDisplayOtherDrawer : ControlDrawer
<ConditionalDisplayOtherControl>

Description

Constructors

ConditionalDisplayOtherDrawer()

   <<<

ConditionalDisplaySelfControl

PiRhoSoft.UtilityEditor.ConditionalDisplaySelfControl : ConditionalDisplayControl

Description

Constructors

ConditionalDisplaySelfControl()

   # Public Methods

float GetHeight(SerializedProperty *property*, **GUIContent** *label***)** *(virtual)*

void Draw(Rect *position*, **SerializedProperty** *property*, **GUIContent** *label***)** *(virtual)*

   <<<

ConditionalDisplaySelfDrawer

PiRhoSoft.UtilityEditor.ConditionalDisplaySelfDrawer : ControlDrawer
<ConditionalDisplaySelfControl>

Description

Constructors

ConditionalDisplaySelfDrawer()

   <<<

ContextMarginScope

PiRhoSoft.UtilityEditor.ContextMarginScope : Scope

Description

Constructors

ContextMarginScope(float *margin***)**
> # Protected Methods

void CloseScope() *(virtual)*
> <<<

ControlDrawer

PiRhoSoft.UtilityEditor.ControlDrawer<*ControlType*> : PropertyDrawer

Description

Constructors

ControlDrawer()
> # Public Methods

float GetPropertyHeight(SerializedProperty *property*, **GUIContent** *label***)** *(virtual)*
void OnGUI(Rect *position*, **SerializedProperty** *property*, **GUIContent** *label***)** *(virtual)*
> <<<

CreateNamedPopup

PiRhoSoft.UtilityEditor.CreateNamedPopup : CreatePopup

Description

Constructors

CreateNamedPopup()
> # Public Fields

bool *IsNameValid*
> # Public Properties

string *Name (read only)*
> # Protected Methods

float GetContentHeight() *(virtual)*

bool DrawContent() *(virtual)*

void Reset() *(virtual)*

    <<<

CreatePopup

PiRhoSoft.UtilityEditor.CreatePopup : PopupWindowContent

Description

Static Fields

float *DefaultWidth*

    # Constructors

CreatePopup()

    # Public Methods

void Setup(GUIContent *label,* **Action** *creator,* **Func<bool>** *validator***)** *(virtual)*

Vector2 GetWindowSize() *(virtual)*

void OnGUI(Rect *rect***)** *(virtual)*

    # Protected Properties

bool *IsValid (read only)*

bool *HasChanged (read only)*

    # Protected Methods

float GetContentHeight() *(virtual)*

bool DrawContent() *(virtual)*

void Reset() *(virtual)*

    <<<

DictionaryControl

PiRhoSoft.UtilityEditor.DictionaryControl : ListControl

Description

Constructors

DictionaryControl()

    # Public Methods

DictionaryControl **Setup(SerializedProperty** *property*, **IEditableDictionary** *dictionary***)**

DictionaryControl **MakeDrawable(ListItemDisplayType** *itemDisplay***)**

DictionaryControl **MakeDrawable(Action<Rect, Rect, Rect, Rect>** *callback***)**

DictionaryControl **MakeAddable(Button** *button*, **GUIContent** *label*, **Action<IEditableDictionary,
IEditableDictionary>** *callback***)**

DictionaryControl **MakeRemovable(Button** *button*, **Action<IEditableDictionary,
IEditableDictionary>** *callback***)**

DictionaryControl **MakeEditable(Button** *button*, **Action<IEditableDictionary,
IEditableDictionary>** *callback***)**

float GetItemHeight(int *index***)**

void DoDefaultDraw(Rect *rect*, **string** *key*, **int** *index***)**

   # Protected Methods

void Draw(Rect *rect*, **int** *index***)** *(virtual)*

   <<<

DictionaryDisplayControl

PiRhoSoft.UtilityEditor.DictionaryDisplayControl : PropertyControl

Description

Static Fields

string *_invalidTypeWarning*

   # Constructors

DictionaryDisplayControl()

   # Public Methods

void Setup(SerializedProperty *property*, **FieldInfo** *fieldInfo*, **PropertyAttribute** *attribute***)**
(virtual)

float GetHeight(SerializedProperty *property*, **GUIContent** *label***)** *(virtual)*

void Draw(Rect *position*, **SerializedProperty** *property*, **GUIContent** *label***)** *(virtual)*

   <<<

DictionaryDisplayDrawer

PiRhoSoft.UtilityEditor.DictionaryDisplayDrawer : ControlDrawer<DictionaryDisplayControl>

Description

Constructors

DictionaryDisplayDrawer()

    <<<

DisableInInspectorControl

PiRhoSoft.UtilityEditor.DisableInInspectorControl : PropertyScopeControl

Description

Constructors

DisableInInspectorControl()

    # Public Methods

float GetHeight(SerializedProperty *property***, GUIContent** *label***)** *(virtual)*
void Draw(Rect *position***, SerializedProperty** *property***, GUIContent** *label***)** *(virtual)*

    <<<

DisableInInspectorDrawer

PiRhoSoft.UtilityEditor.DisableInInspectorDrawer : ControlDrawer<DisableInInspectorControl>

Description

Constructors

DisableInInspectorDrawer()

    <<<

EditObjectScope

PiRhoSoft.UtilityEditor.EditObjectScope : Scope

Description

Constructors

EditObjectScope(SerializedObject *serializedObject***)**

    # Protected Methods

void CloseScope() *(virtual)*

    <<<

EnterField

PiRhoSoft.UtilityEditor.EnterField

Description

Static Methods

bool DrawInt(string *controlName*, **GUIContent** *label*, **int** *value (ref)***)**

bool DrawInt(string *controlName*, **Rect** *position*, **GUIContent** *label*, **int** *value (ref)***)**

bool DrawFloat(string *controlName*, **GUIContent** *label*, **float** *value (ref)***)**

bool DrawFloat(string *controlName*, **Rect** *position*, **GUIContent** *label*, **float** *value (ref)***)**

bool DrawString(string *controlName*, **GUIContent** *label*, **string** *text (ref)***)**

bool DrawString(string *controlName*, **Rect** *position*, **GUIContent** *label*, **string** *text (ref)***)**

   # Constructors

EnterField()

   <<<

EnumDisplayDrawer

PiRhoSoft.UtilityEditor.EnumDisplayDrawer : PropertyDrawer

Description

Static Methods

void Draw(GUIContent *label*, **SerializedProperty** *property*, **Type** *enumType*, **EnumDisplayType** *type*, **bool** *forceFlags*, **float** *minimumButtonWidth***)**

void Draw(Rect *position*, **GUIContent** *label*, **SerializedProperty** *property*, **Type** *enumType*, **EnumDisplayType** *type*, **bool** *forceFlags*, **float** *minimumButtonWidth***)**

float GetHeight<EnumType>(GUIContent *label*, **EnumDisplayType** *type*, **float** *minimumButtonWidth***)**

int Draw<EnumType>(GUIContent *label*, **int** *value*, **EnumDisplayType** *type*, **bool** *forceFlags*, **float** *minimumButtonWidth***)**

int Draw<EnumType>(Rect *position*, **GUIContent** *label*, **int** *value*, **EnumDisplayType** *type*, **bool** *forceFlags*, **float** *minimumButtonWidth***)**

float GetHeight(GUIContent *label*, **Type** *enumType*, **EnumDisplayType** *type*, **float** *minimumButtonWidth***)**

int Draw(GUIContent *label*, **int** *value*, **Type** *enumType*, **EnumDisplayType** *type*, **bool** *forceFlags*, **float** *minimumButtonWidth***)**

int Draw(Rect *position*, **GUIContent** *label*, **int** *value*, **Type** *enumType*, **EnumDisplayType** *type*, **bool** *forceFlags*, **float** *minimumButtonWidth***)**

   # Constructors

EnumDisplayDrawer()

Public Methods

float GetPropertyHeight(SerializedProperty *property*, **GUIContent** *label***)** *(virtual)*

void OnGUI(Rect *position*, **SerializedProperty** *property*, **GUIContent** *label***)** *(virtual)*

    <<<

FloatPreference

PiRhoSoft.UtilityEditor.FloatPreference

Description

Constructors

FloatPreference(string *name*, **float** *defaultValue***)**

Public Properties

float *Value*

    <<<

HandleColorScope

PiRhoSoft.UtilityEditor.HandleColorScope : Scope

Description

Constructors

HandleColorScope()

HandleColorScope(Color *color***)**

Protected Methods

void CloseScope() *(virtual)*

    <<<

HandleHelper

PiRhoSoft.UtilityEditor.HandleHelper

Description

Static Methods

void DrawText(Vector2 *position*, **string** *text*, **TextAnchor** *alignment*, **Color** *color***)**

void DrawArrow(Vector2 *position*, **Vector2** *direction*, **float** *length*, **Color** *color***)**

void DrawLine(Vector2 *start*, **Vector2** *end*, **Color** *color***)**

void DrawCircle(Vector2 *position*, **float** *radius*, **Color** *color***)**

void DrawBezier(Vector2 *start*, **Vector2** *end*, **Color** *color***)**

Rect BoundsHandle(Rect *bounds*, **Vector2** *snap*, **Color** *rectangleOutline*, **Color** *rectangleFill*, **Color** *circleOutline*, **Color** *circleFill*, **float** *handleSize***)**

Vector2 MoveHandle(Vector2 *position*, **Vector2** *size*, **Vector2** *snap*, **Color** *outline*, **Color** *fill***)**

Vector2 MoveHandle(Vector2 *position*, **Vector2** *snap*, **Color** *outline*, **Color** *fill*, **float** *handleSize***)**

Rect ScaleHandles(Rect *bounds*, **Vector2** *snap*, **Color** *outline*, **Color** *fill*, **float** *handleSize***)**

    <<<

Icon

PiRhoSoft.UtilityEditor.Icon

Description

Static Fields

string *Add*
string *CustomAdd*
string *Remove*
string *Edit*
string *Expanded*
string *Collapsed*
string *Refresh*
string *Load*
string *Unload*

    # Static Methods

Icon BuiltIn(string *name***)**

Icon Base64(string *data***)**

    # Constructors

Icon()

    # Public Properties

Texture *Content (read only)*

    <<<

IntPreference

PiRhoSoft.UtilityEditor.IntPreference

Description

Constructors

IntPreference(string *name*, **int** *defaultValue*)

   # Public Properties

int *Value*

   <<<

InvalidScope

PiRhoSoft.UtilityEditor.InvalidScope : ColorScope

Description

Static Fields

Color *ValidBackground*

Color *InvalidBackground*

   # Constructors

InvalidScope(bool *valid*)

   <<<

JsonPreference

PiRhoSoft.UtilityEditor.JsonPreference*<T>*

Description

Constructors

JsonPreference(string *name*)

   # Public Properties

T *Value*

   <<<

Label

PiRhoSoft.UtilityEditor.Label

Description

Static Methods

string GetTooltip(Type *type***, string** *propertyName***)**
string GetTooltip(FieldInfo *field***)**
    # Constructors

Label(Type *type***, string** *property***)**
    # Public Properties

GUIContent *Content (read only)*
    <<<

LabelWidthScope

PiRhoSoft.UtilityEditor.LabelWidthScope : Scope

Description

Constructors

LabelWidthScope(float *width***)**
    # Protected Methods

void CloseScope() *(virtual)*
    <<<

ListControl

PiRhoSoft.UtilityEditor.ListControl

Description

Static Fields

float *HeaderHeight*
float *CollapsedHeight*
float *ItemDefaultHeight*
float *ItemPadding*

float *TotalMargin*

Public Properties

ReorderableList *List (read only)*

bool *Visible*

Public Methods

ListControl **MakeCollapsable(string** *preferenceName***)**

ListControl **MakeEmptyLabel(GUIContent** *label***)**

ListControl **MakeReorderable(Action<int, int>** *callback***)**

ListControl **MakeHeaderButton(Button** *button***, Action<Rect>** *callback***, Color** *color***)**

ListControl **MakeHeaderButton(Button** *button***, GenericMenu** *menu***, Color** *color***)**

ListControl **MakeHeaderButton(Button** *button***, PopupWindowContent** *popup***, Color** *color***)**

ListControl **MakeItemButton(Button** *button***, Action<Rect, Rect>** *callback***, Color** *color***)**

ListControl **MakeCustomHeight(ElementHeightCallbackDelegate** *callback***)**

float GetHeight()

void Draw(GUIContent *label***)**

void Draw(Rect *position***, GUIContent** *label***)**

Protected Constructors

ListControl()

Protected Methods

void Draw(Rect *rect***, int** *index***)** *(abstract)*

void Setup(ReorderableList *list***)**

void ElementsMoved(ReorderableList *list***, int** *oldIndex***, int** *newIndex***)** *(virtual)*

<<<

ListDisplayControl

PiRhoSoft.UtilityEditor.ListDisplayControl : PropertyControl

Description

Constructors

ListDisplayControl()

Public Methods

void Setup(SerializedProperty *property***, FieldInfo** *fieldInfo***, PropertyAttribute** *attribute***)**
(virtual)

float GetHeight(SerializedProperty *property***, GUIContent** *label***)** *(virtual)*

void Draw(Rect *position***, SerializedProperty** *property***, GUIContent** *label***)** *(virtual)*

<<<

ListDisplayDrawer

PiRhoSoft.UtilityEditor.ListDisplayDrawer : ControlDrawer<ListDisplayControl>

Description

Constructors

ListDisplayDrawer()

    <<<

ListProxy

PiRhoSoft.UtilityEditor.ListProxy : ICollection, IEnumerable, IList

Description

Public Properties

int *Count (read only) (abstract)*
Object *Item (abstract)*
bool *IsFixedSize (read only) (virtual)*
bool *IsReadOnly (read only) (virtual)*
bool *IsSynchronized (read only) (virtual)*
Object *SyncRoot (read only) (virtual)*

    # Public Methods

int Add(Object *value***)** *(virtual)*
void Clear() *(virtual)*
bool Contains(Object *value***)** *(virtual)*
void CopyTo(Array *array***, int** *index***)** *(virtual)*
IEnumerator GetEnumerator() *(virtual)*
int IndexOf(Object *value***)** *(virtual)*
void Insert(int *index***, Object** *value***)** *(virtual)*
void Remove(Object *value***)** *(virtual)*
void RemoveAt(int *index***)** *(virtual)*

    # Protected Constructors

ListProxy()

    <<<

MaximumControl

PiRhoSoft.UtilityEditor.MaximumControl : PropertyScopeControl

Description

Static Fields

string *_invalidTypeWarning*
Constructors

MaximumControl()
Public Methods

void Setup(SerializedProperty *property***, FieldInfo** *fieldInfo***, PropertyAttribute** *attribute***)**
(virtual)
float GetHeight(SerializedProperty *property***, GUIContent** *label***)** *(virtual)*
void Draw(Rect *position***, SerializedProperty** *property***, GUIContent** *label***)** *(virtual)*
<<<

MaximumDrawer

PiRhoSoft.UtilityEditor.MaximumDrawer : ControlDrawer<MaximumControl>

Description

Constructors

MaximumDrawer()
<<<

MinimumControl

PiRhoSoft.UtilityEditor.MinimumControl : PropertyScopeControl

Description

Static Fields

string *_invalidTypeWarning*
Constructors

MinimumControl()
Public Methods

void Setup(SerializedProperty *property***, FieldInfo** *fieldInfo***, PropertyAttribute** *attribute***)**
(virtual)

float GetHeight(SerializedProperty *property***, GUIContent** *label***)** *(virtual)*

void Draw(Rect *position***, SerializedProperty** *property***, GUIContent** *label***)** *(virtual)*

    <<<

MinimumDrawer

PiRhoSoft.UtilityEditor.MinimumDrawer : ControlDrawer<MinimumControl>

Description

Constructors

MinimumDrawer()

    <<<

ObjectControl

PiRhoSoft.UtilityEditor.ObjectControl<*T*> : PropertyControl

Description

Public Methods

void Setup(T *target***, SerializedProperty** *property***, FieldInfo** *fieldInfo***, PropertyAttribute**
*attribute***)** *(abstract)*

float GetHeight(GUIContent *label***)** *(abstract)*

void Draw(Rect *position***, GUIContent** *label***)** *(abstract)*

void Setup(SerializedProperty *property***, FieldInfo** *fieldInfo***, PropertyAttribute** *attribute***)**
(virtual)

float GetHeight(SerializedProperty *property***, GUIContent** *label***)** *(virtual)*

void Draw(Rect *position***, SerializedProperty** *property***, GUIContent** *label***)** *(virtual)*

void Draw(GUIContent *label***)**

    # Protected Constructors

ObjectControl()

    <<<

ObjectListControl

PiRhoSoft.UtilityEditor.ObjectListControl : ListControl

Description

Constructors

ObjectListControl()

   # Public Methods

ObjectListControl Setup(IList *list***)**

ObjectListControl MakeDrawable(Action<Rect, Rect, Rect> *callback***)**

ObjectListControl MakeAddable(Button *button***, Action<IList>** *callback***)**

ObjectListControl MakeRemovable(Button *button***, Action<IList, IList>** *callback***)**

ObjectListControl MakeEditable(Button *button***, Action<IList, IList>** *callback***)**

void DoDefaultAdd()

void DoDefaultEdit(int *index***)**

void DoDefaultRemove(int *index***)**

   # Protected Methods

void Draw(Rect *rect***, int** *index***)** *(virtual)*

   <<<

PropertyControl

PiRhoSoft.UtilityEditor.PropertyControl

Description

Public Methods

void Setup(SerializedProperty *property***, FieldInfo** *fieldInfo***, PropertyAttribute** *attribute***)** *(abstract)*

float GetHeight(SerializedProperty *property***, GUIContent** *label***)** *(abstract)*

void Draw(Rect *position***, SerializedProperty** *property***, GUIContent** *label***)** *(abstract)*

   # Protected Constructors

PropertyControl()

   <<<

PropertyHelper

PiRhoSoft.UtilityEditor.PropertyHelper

Description

Static Methods

SerializedProperty GetSibling(**SerializedProperty** *property*, **string** *siblingName*)

T GetObject<T>(**SerializedProperty** *property*)

   <<<

PropertyListControl

PiRhoSoft.UtilityEditor.PropertyListControl : ListControl

Description

Constructors

PropertyListControl()

   # Public Methods

PropertyListControl Setup(**SerializedProperty** *property*)

PropertyListControl MakeDrawable(**ListItemDisplayType** *itemDisplay*)

PropertyListControl MakeDrawable(**Action**<**Rect**, **Rect**, **Rect**> *callback*)

PropertyListControl MakeAddable(**Button** *button*, **Action**<**SerializedProperty**> *callback*)

PropertyListControl MakeRemovable(**Button** *button*, **Action**<**SerializedProperty**, **SerializedProperty**> *callback*)

PropertyListControl MakeEditable(**Button** *button*, **Action**<**SerializedProperty**, **SerializedProperty**> *callback*)

void DoDefaultAdd()

void DoDefaultEdit(int *index*)

void DoDefaultRemove(int *index*)

float GetItemHeight(int *index*)

void DoDefaultDraw(Rect *rect*, **int** *index*)

   # Protected Methods

void Draw(Rect *rect*, **int** *index*) *(virtual)*

void ElementsMoved(ReorderableList *list*, **int** *oldIndex*, **int** *newIndex*) *(virtual)*

   <<<

PropertyScopeControl

PiRhoSoft.UtilityEditor.PropertyScopeControl : PropertyControl

Description

Static Methods

PropertyDrawer **GetNextDrawer(FieldInfo** *fieldInfo*, **PropertyAttribute** *attribute***)**

    # Public Methods

void Setup(SerializedProperty *property*, **FieldInfo** *fieldInfo*, **PropertyAttribute** *attribute***)** *(virtual)*

    # Protected Constructors

PropertyScopeControl()

    # Protected Methods

float GetNextHeight(SerializedProperty *property*, **GUIContent** *label***)**

void DrawNext(Rect *position*, **SerializedProperty** *property*, **GUIContent** *label***)**

    <<<

RectHelper

PiRhoSoft.UtilityEditor.RectHelper

Description

Static Fields

float *VerticalSpace*

float *HorizontalSpace*

float *IconWidth*

float *LineHeight*

float *LeftMargin*

float *RightMargin*

    # Static Properties

float *CurrentIndentWidth (read only)*

float *CurrentLabelWidth (read only)*

float *CurrentFieldWidth (read only)*

float *CurrentViewWidth (read only)*

float *CurrentContextWidth (read only)*

float *ContextMargin*

    # Static Methods

float GetLabeledWidth(GUIContent *label***)**

Rect TakeLine(Rect *full (ref)***)**

Rect TakeHeight(Rect *full (ref)***, float** *height***)**

Rect TakeTrailingHeight(Rect *full (ref)***, float** *height***)**

Rect TakeVerticalSpace(Rect *full (ref)***)**

Rect TakeHorizontalSpace(Rect *full (ref)***)**

void TakeIndent(Rect *full (ref)***)**

Rect TakeLabel(Rect *full (ref)***)**

Rect TakeLeadingIcon(Rect *full (ref)***)**

Rect TakeTrailingIcon(Rect *full (ref)***)**

Rect TakeWidth(Rect *full (ref)***, float** *width***)**

Rect TakeTrailingWidth(Rect *full (ref)***, float** *width***)**

Rect Inset(Rect *rect***, float** *padding***)**

Rect Inset(Rect *rect***, float** *left***, float** *right***, float** *top***, float** *bottom***)**

Rect Adjust(Rect *rect***, float** *width***, float** *height***, RectHorizontalAlignment** *horizontal***, RectVerticalAlignment** *vertical***)**

Rect AdjustHeight(Rect *rect***, float** *height***, RectVerticalAlignment** *alignment***)**

Rect AdjustWidth(Rect *rect***, float** *width***, RectHorizontalAlignment** *alignment***)**

float GetIndentWidth(int *levels***)**

   <<<

RectHorizontalAlignment

PiRhoSoft.UtilityEditor.RectHorizontalAlignment

Description

Values

RectHorizontalAlignment *Left*

RectHorizontalAlignment *Center*

RectHorizontalAlignment *Right*

   <<<

RectVerticalAlignment

PiRhoSoft.UtilityEditor.RectVerticalAlignment

Description

Values

RectVerticalAlignment *Top*
RectVerticalAlignment *Middle*
RectVerticalAlignment *Bottom*
    <<<

ReloadOnChangeControl

PiRhoSoft.UtilityEditor.ReloadOnChangeControl : PropertyScopeControl

Description

Constructors

ReloadOnChangeControl()
   # Public Methods

float GetHeight(SerializedProperty *property*, **GUIContent** *label*) *(virtual)*
void Draw(Rect *position*, **SerializedProperty** *property*, **GUIContent** *label*) *(virtual)*
    <<<

ReloadOnChangeDrawer

PiRhoSoft.UtilityEditor.ReloadOnChangeDrawer : ControlDrawer<ReloadOnChangeControl>

Description

Constructors

ReloadOnChangeDrawer()
    <<<

SceneData

PiRhoSoft.UtilityEditor.SceneData : ValueType

Description

Public Fields

bool *IsActive*
bool *IsLoaded*
string *Path*

SceneHelper

PiRhoSoft.UtilityEditor.SceneHelper : AssetPostprocessor

Description

Static Methods

void RefreshLists()

SceneList **GetSceneList(bool** *includeNone*, **bool** *includeCreate*)

Scene **CreateScene(string** *name*, **Action** *create*)

void AddSceneToBuild(Scene *scene*)

SceneState **CaptureState()**

void RestoreState(SceneState *state*)

   # Constructors

SceneHelper()

   <<<

SceneList

PiRhoSoft.UtilityEditor.SceneList

Description

Constructors

SceneList()

   # Public Fields

bool *HasNone*

bool *HasCreate*

GUIContent[] *Names*

List<**string**> *Paths*

   # Public Properties

int *CreateIndex (read only)*

   # Public Methods

int GetIndex(string *path*)

string GetPath(int *index*)

   <<<

SceneState

PiRhoSoft.UtilityEditor.SceneState

Description

Constructors

SceneState()
 # Public Fields

SceneData[] *Scenes*

 <<<

SnapControl

PiRhoSoft.UtilityEditor.SnapControl : PropertyScopeControl

Description

Constructors

SnapControl()
 # Public Methods

void Setup(SerializedProperty *property*, **FieldInfo** *fieldInfo*, **PropertyAttribute** *attribute***)** *(virtual)*
float GetHeight(SerializedProperty *property*, **GUIContent** *label***)** *(virtual)*
void Draw(Rect *position*, **SerializedProperty** *property*, **GUIContent** *label***)** *(virtual)*

 <<<

SnapDrawer

PiRhoSoft.UtilityEditor.SnapDrawer : ControlDrawer<SnapControl>

Description

Constructors

SnapDrawer()

 <<<

StringPreference

PiRhoSoft.UtilityEditor.StringPreference

Description

Constructors

StringPreference(string *name*, **string** *defaultValue*)

    # Public Properties

string *Value*

    <<<

Style

PiRhoSoft.UtilityEditor.Style

Description

Constructors

Style(Func<**GUIStyle**> *create*)

    # Public Properties

GUIStyle *Content (read only)*

    <<<

TypeHelper

PiRhoSoft.UtilityEditor.TypeHelper

Description

Static Fields

List<**Type**> *SerializableTypes*

    # Static Methods

bool HasAttribute<AttributeType>(Type *type***)**

AttributeType GetAttribute<AttributeType>(Type *type***)**

AttributeType GetAttribute<AttributeType>(FieldInfo *field***)**

bool HasAttribute(Type *type***, Type** *attributeType***)**

Attribute GetAttribute(Type *type***, Type** *attributeType***)**

Attribute GetAttribute(FieldInfo *field***, Type** *attributeType***)**

T CreateInstance<T>(Type *type***)**

bool IsCreatableAs<BaseType>(Type *type***)**

bool IsCreatableAs(Type *baseType***, Type** *type***)**

List<Type> ListDerivedTypes<BaseType>()

List<Type> ListDerivedTypes(Type *baseType***)**

List<Type> ListTypesWithAttribute<AttributeType>()

List<Type> ListTypesWithAttribute(Type *attributeType***)**

IEnumerable<Type> FindTypes(Func<Type, Type> *predicate***)**

TypeList GetTypeList<T>(bool *includeNone***)**

TypeList GetTypeList(Type *baseType***, bool** *includeNone***)**

bool IsSerializable(FieldInfo *field***)**

bool IsSerializable(Type *type***)**

    <<<

TypeList

PiRhoSoft.UtilityEditor.TypeList

Description

Constructors

TypeList()

    # Public Fields

Type *BaseType*

bool *HasNone*

GUIContent[] *Names*

List<Type> *Types*

    # Public Methods

int GetIndex(Type *type***)**

Type GetType(int *index***)**

    <<<

TypePopupDrawer

PiRhoSoft.UtilityEditor.TypePopupDrawer

Description

Static Methods

float GetHeight()

Type Draw<BaseType>(GUIContent *label*, **Type** *type*)

Type Draw<BaseType>(Rect *position*, **GUIContent** *label*, **Type** *type*)

   <<<

UndoScope

PiRhoSoft.UtilityEditor.UndoScope : Scope

Description

Constructors

UndoScope(Object *objectToTrack*, **bool** *forceDirty*)

UndoScope(SerializedObject *serializedObject*)

   # Protected Methods

void CloseScope() *(virtual)*