

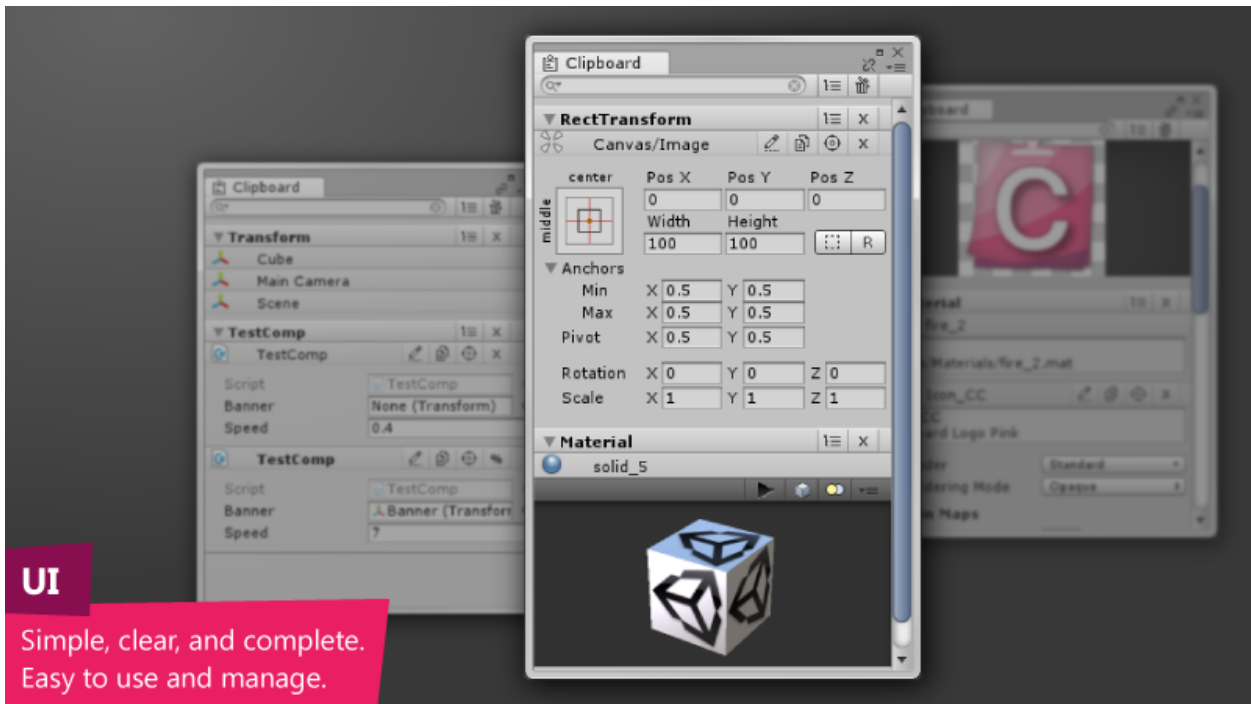
# CLIPBOARD

Let's save your time from routine work.

- Have you ever tweaked lots of Components in play mode, and noted values for later?
- Do you need to try many settings or Material properties to pick which looks better?
- Do you even want to copy a lot, then paste after lunch or the weekend?
- Would you like a Component presets panel to apply values quickly?

That's why I made this!

The videos below show the **brief** and the **smooth workflow**, and how to use as a **presets panel**.



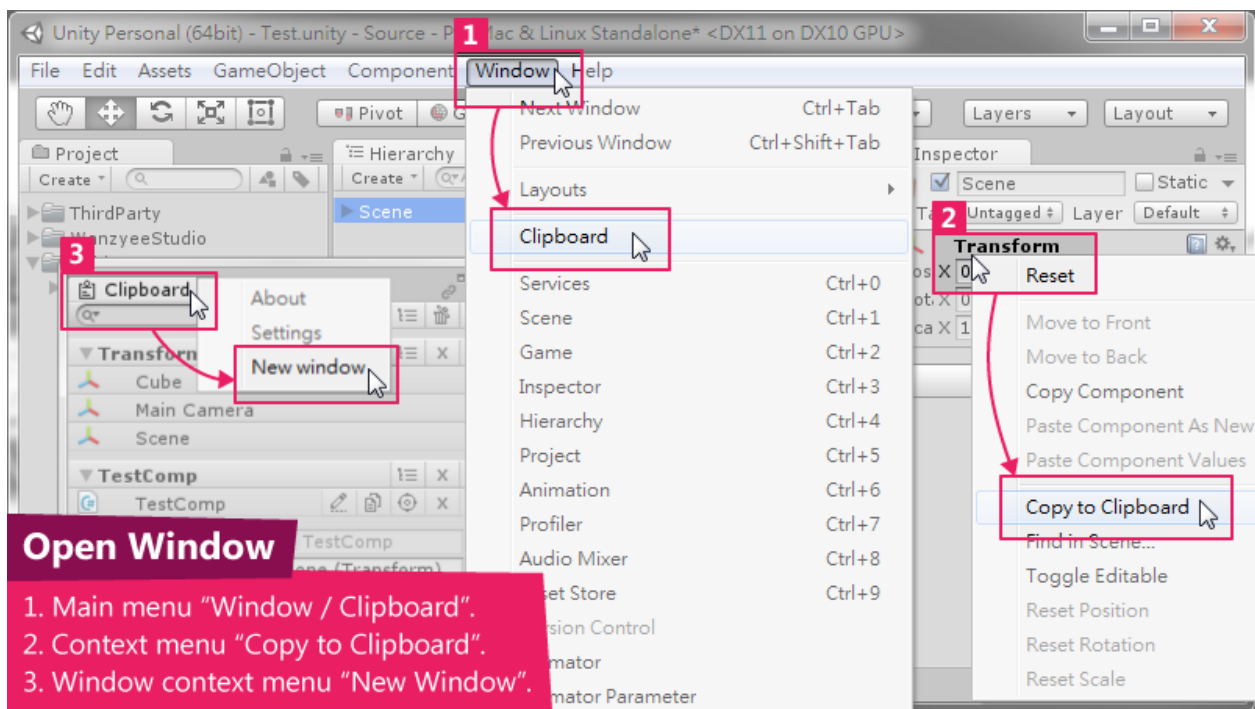
## Overview:

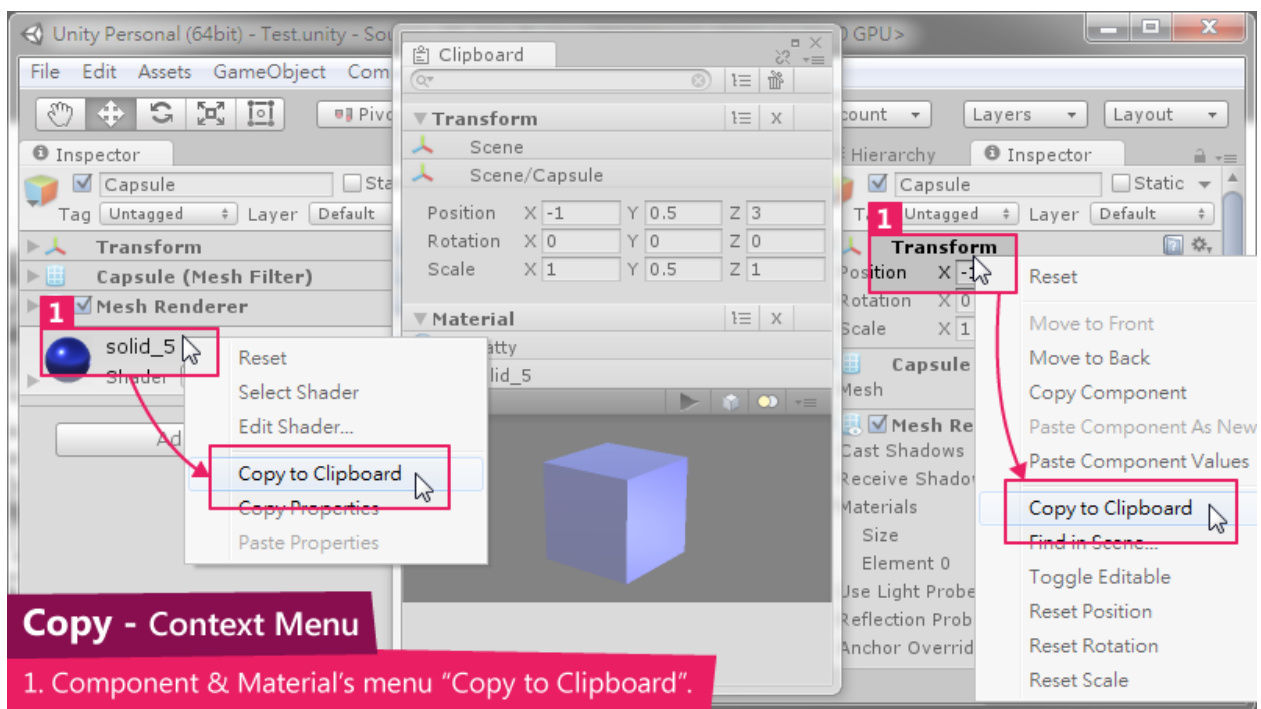
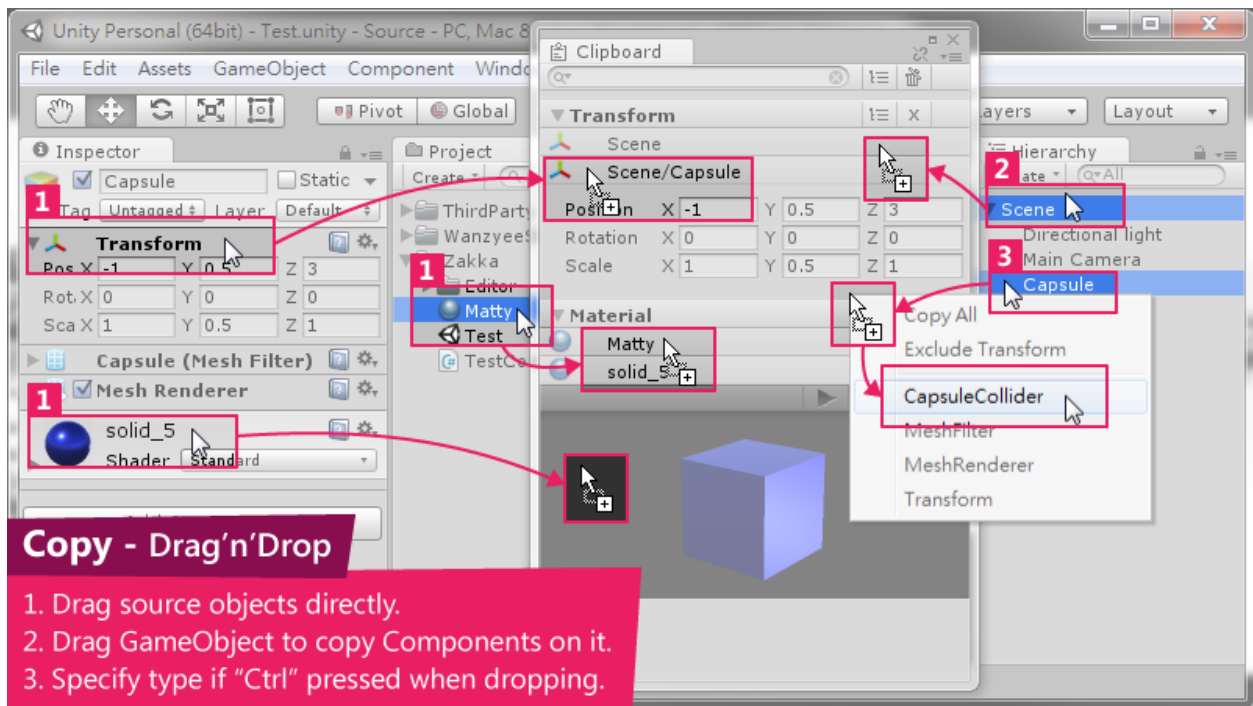
- Easy to copy and paste multiple Components or Materials.
- Just drag'n'drop to snapshot them, and drag'n'drop to paste the copies back.
- All the copies can be managed and edited in a foldable Inspector-like window.
- With custom labels and quick search bar, it could work as a presets panel.
- One click to find the copy source.
- Most copies will be saved in the project.

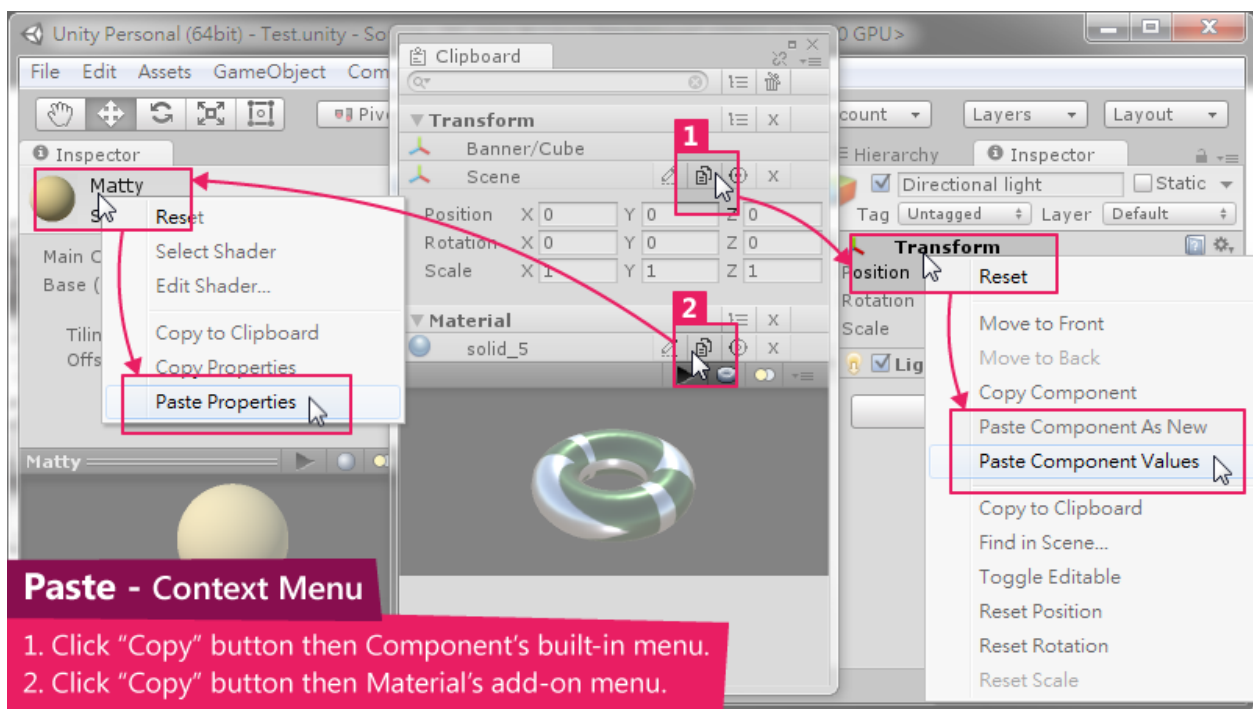
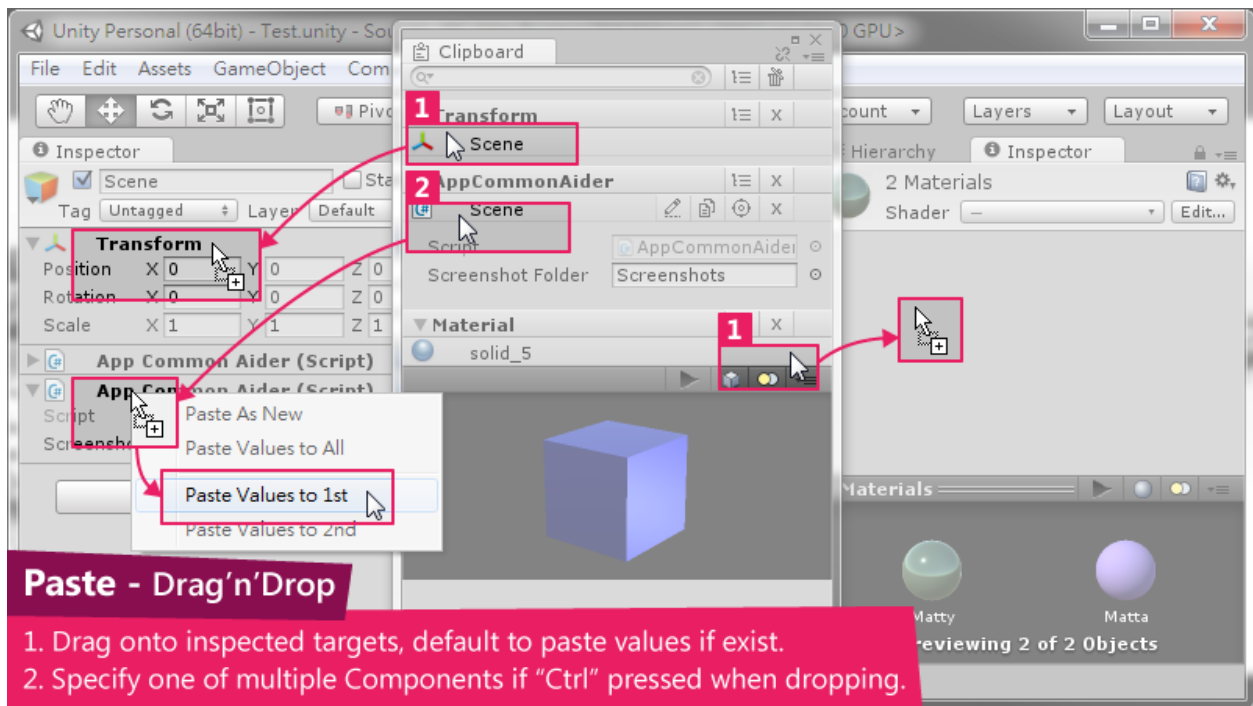
The further technical documentation is available [here](#).

And the operation manual is right below.

## Using:







**UI - Manage**

Convenient functions.  
Hover to show buttons.

- Context menu to set if prefer persisting.
- Fold or expand all.
- Delete all.
- Delete the group.
- Fold or expand all items.
- Fold or expand the group.
- Delete the item.
- Fold or expand the item.
- Bold label indicates it's not persistent.
- Make persistent without scene references.
- Click to ping the copy source.
- Double click to select it.
- Click to copy to buffer for menu usage.
- Double click to duplicate the item.

**UI - Search**

Filter items with a search bar.  
Identify with custom labels.

- Quick search menu.
- Auto parse current item keywords.
- Needless to key in manually.
- Auto filter by selected objects.
- Search bar.
- Any text filters by item name.
- Prefix "t:" by type and "l:" by label.
- Default label indicates the copy source.
- Edit the item label.
- The 1st line is the item name.
- The others from the 2nd line are labels.

### Add-on:

- Here's **AuxCursor** to show cross windows dragging hint, useful for editor developer.
- This also includes a small utility: **Material Property Copier**.
- It provides context menus **"Copy Properties"** and **"Paste Properties"** for Material.
- It works just like the similar menus of a Component.

## Release Notes:

### 1.2.5

- New, window context menu "Settings" to toggle if prefer persisting.
- Fix, prevent errors caused by missing components.
- Compile Lite version and upload by the same Unity version to avoid plugin settings wrong.

### 1.2.4

- Fix, a mouse up aborting exception may occurs when drop to open the paste menu.

### 1.2.3

- Fix, quick search menu to determine the existing keywords.
- Change the package name.

### 1.2.2

- Free version, drag'n'drop Transform, Camera, Light to copy and paste for trial.
- Free version, able to filter items with the search bar.
- Fix, error occurs when use the context menu to copy more than the limit in free version.
- Fix, API "track" didn't work properly.
- Fix, unfocus the text field when change layout to ensure display correctly.
- Fix, potential uncleared editor error.
- Add the label "Ignore" to the asset copies in case.

### 1.2.1

- New, double-click the Copy button to duplicate a copied item.

### 1.2

- More features for free version, e.g., more copies, fold item, preview, find source, API.
- New, custom item label for search and identification.
- New, quick search menu to select filters without manual typing.
- New, auto filter by tracking selected objects.
- Change the search bar to work like the Project window's instead of regex.
- Keep the search filter after exiting the editor.
- Keep the fold states after exiting the editor.
- Clearer GUI, show item's buttons only when mouse hover, and adjust layout a little.
- Show a confirm dialog when click "Clear all" button to avoid miss operation.
- Save label data to AssetImporter.userData instead of a temp file in assets.
- Optimize, lower reload frequency with a new trigger system, obvious while playing.
- Optimize, new faster reload system for multiple windows usage.
- Change the store folder from "Temps/Clipboard" to "Temp/Clipboard".
- Change EditrixStyle API from "Create" to "Load", and reuse the generated texture.
- Fix, drag an object into a field of a copied item.
- Fix, Inspector reflection for dragging paste in Unity 5.5.0p3.
- Fix, support to copy audio filters.
- Improve compatibility for the same name types of different assemblies.

### 1.1.7

- Move the toolbar "Window" and "Tooltip" buttons to the window context menu.
- Change the toolbar "Clear" button icon instead of text.

### 1.1.6

- Fix the random crash caused by removing component in Unity 5.5.
- Fix the hideFlags error occurs when copying component in Unity 5.5.

### 1.1.5

- Add the documentation link to the window context menu "Help".
- Remove the big tooltip shows when there's no copied items.

#### 1.1.4

- Change the drag'n'drop pasting system to avoid some weird behavior on Mac.
- It's able to cancel dragging paste by press "Esc" key.

#### 1.1.3

- Fix showing the Save Assets dialog when copying, if the Verify Saving Assets flag set.
- Fix the null reference of an unexpected empty prefab when loading copies.
- Fix creating the style images from the window constructor.

#### 1.1.2

- Fix an error when cleaning invalid Editor by checking its target.

#### 1.1.1

- Add a regex search bar to filter labels.
- Rename the dll and move to an Editor folder to avoid exporting when building player.
- It's recommended to remove the old package before updating.

#### 1.1

- New, Drag'n'Drop to paste.
- New, able to copy all components on a GameObject by dragging it.
- Fix copying a component which isn't a specific type, e.g., Halo is a Behaviour.

#### 1.0.5

- Improve the source finding function to find in an unsaved scene.
- Fix the obsolete API about the HSV color used for icons.
- Fix the reference to the copy source when a copied component references to itself.
- Fix component copy operation to be not undoable since it shouldn't be.

#### 1.0.4

- Fix the copied asset isn't saved caused by the previous bug fixing for weird object.
- Fix copying a material which inherits from Material in case.
- Fix the reference lost when a material refers to a non-asset texture.

#### 1.0.3

- Add a dark background to every preview to make UI identical.
- Add a public method to open window, for users to make their own hotkey.
- Fix copying a component which inherits from Transform, e.g., RectTransform.
- Fix a weird object shown in Hierarchy when copy a component related to RectTransform.

#### 1.0.2

- New, a button to easily find copy source object, and adjust the UI for it.
- Fix material copy operation to ensure be saved.

#### 1.0.1

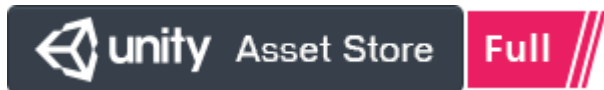
- Fix the cursor flashing when the Project or Hierarchy window changes constantly.
- Fix the component editor usage to avoid crash when a copied asset is operated manually.

#### 1.0

- First release.

### Publish Notes:

- Requires: Unity 5.3.5f1 or higher.
- Category: Editor Extensions / Utilities
- Keywords: Component Material Clipboard Copy Paste Snap Snapshot Drag Drop Editor Extension Utility Inspector Custom Preset Pool Search Filter



### Full-only Features:

- Unlimited copy numbers. (10 for free version)
- Keep the copies after exit editor.
- Full Drag'n'Drop operations. (Transform, Camera, Light for free version)
- Quick search menu.
- Auto filter by tracking selection.
- Custom item label.
- Open multiple windows.



# Clipboard

[Released Packages](#) » [ClipboardSpecific Classes](#) » |  
[Editor Extensions](#)

Utility to copy `UnityEngine.Component` or `UnityEngine.Material` and paste it back. [More...](#)

Inherits `EditorWindow`, and `IHasCustomMenu`.

## Public Member Functions

void **AddItemsToMenu** (GenericMenu menu)  
Add the window context menu items. [More...](#)

## Static Public Member Functions

static void **OpenWindow** ()  
Show the clipboard window. [More...](#)

static bool **IsCopyable** (Object source)  
Determine if the specified source is able to copy to clipboard. [More...](#)

static void **Copy** (params Object[] sources)  
Copy the specified sources to clipboard. [More...](#)

static void **Clear** (Type type=null)  
Clear clipboard by specified type, or pass null to clear all. [More...](#)

## Public Attributes

bool **track**  
Flag to track selection to change search filter automatically. [More...](#)

string **search** = ""  
The search filter pattern in the search bar. [More...](#)

bool **setting**  
Flag to show settings. [More...](#)

## Properties

static bool **preferPersist** [get, set]  
Flag to persist new copies as asset if possible. [More...](#)

## Detailed Description

Utility to copy `UnityEngine.Component` or `UnityEngine.Material` and paste it back.

Useful for tweaking lots of objects in the editor, even in play mode. Click the menu "Window/Clipboard" to open the window. Just play and tweak, drag and drop, copy and paste whenever. It acts as sort of preset system, edit

lots of copies as presets, and paste to apply values quickly. Easy to find and manage copies with a search filter, custom item label, and foldable inspector.

Copy:

1. Drag'n'Drop from "Inspector" to copy the inspected sources.
2. Drag'n'Drop a `UnityEngine.GameObject` to copy the components on it.
3. Show menu to specify the component type if "Ctrl" pressed when dropping gameObjects.
4. Or click the context menu "Copy to Clipboard".

Paste:

1. Drag'n'Drop to "Inspector" to paste back to the inspected targets.
2. When dragging a component, it pastes values to the first one on the gameObject, or pastes as new if none.
3. Show menu to specify a target of multiple components if "Ctrl" pressed when dropping.
4. Or click the "Copy" button above any item, and paste by the target's context menu.

Filter items with the search bar:

1. Click the "Magnifier" icon to show a context menu made from the current items to select filters easily.
2. Or input any text to filter the item's name, just like the Project window's search bar.
3. Prefix "t:" filters by the type, the search will include all specified types.
4. Prefix "l:" filters by the tooltip as labels, an item has to match all specified labels.
5. Toggle the "Link" icon at the top-right of window to filter automatically by tracking selection.

Edit the item label:

1. Click the "Pen" button beside the label to show or hide the edit field.
2. To save the change, just press "Ctrl-Enter" keys or unfocus the field after editing.
3. The first line shows as the label title, and the full text is the tooltip.
4. Leave the field empty to show the default text, i.e., the copy source path.

Find the copy source object:

1. Click the "Aim" button to ping it or double-click to select it.
2. The saved trace path is also shown as the default label tooltip.
3. It'll beep if not found, e.g., the trace may be lost if the source is moved or renamed.

The reasons not to save the trace by references below:

1. A scene object reference will change when load a scene.
2. We can't save the scene reference in the project assets.
3. We shouldn't save the edit data in user's game scene.

For component references to scene object. This creates copies and store in editor scene temporarily to ensure content correct. It means the copy will be destroyed when quitting the editor. And also, the scene references will become missing when opening another scene. The situation above is applicable to a material with scene texture, too.

For material or component without reference to any scene object. The copies will be saved with a label "Ignore" in an asset folder to make them still until manually "Clear". This tracks all copies by specific name or folder to

ensure valid after script reloaded.

SVN users may clear all manually to avoid committing, or ignore the storing folder below:

1. The default is "Temp/Clipboard" under the root folder "Assets/WanzyeeStudio".
2. If the root is moved, it becomes "Temp/Clipboard" under the first found "WanzyeeStudio".
3. If there's no "WanzyeeStudio" folder, it'll be "Assets/Temp/Clipboard".

Instructions, to copy and paste generic component is dangerous, even if reflect all the fields. Since we'll never know what the developer do when the component awake. As the [thread](#) I commented, we'd be very careful with which aren't made by ourselves. Finally, this was created, works in the editor with Unity built-in classes and API.

## Member Function Documentation

---

**static void** `OpenWindow ( )`

static

Show the clipboard window.

**static bool** `IsCopyable ( Object source )`

static

Determine if the specified source is able to copy to clipboard.

### Returns

true if is copyable; otherwise, false.

### Parameters

**source** Source object.

**static void** `Copy ( params Object[] sources )`

static

Copy the specified sources to clipboard.

### Parameters

**sources** Source objects.

**static void** `Clear ( Type type = null )`

static

Clear clipboard by specified type, or pass null to clear all.

**void AddItemsToMenu ( GenericMenu menu )**

Add the window context menu items.

Menu "About" to open the online manual, and "Settings" for common preferences. Menu "New window" to open another **Clipboard** window.

**Parameters**

menu Menu.

## Member Data Documentation

---

**bool track**

Flag to track selection to change search filter automatically.

**string search = ""**

The search filter pattern in the search bar.

**bool setting**

Flag to show settings.

## Property Documentation

---

**bool preferPersist**

static get set

Flag to persist new copies as asset if possible.

Enable to save the copies for next time opening the project. Disable to make copying faster without creating asset if you don't need. It only applies to copy operations after changing this setting.

If to persist.

# MaterialPropertyCopier

[Specific Classes](#) » [Editor Extensions](#)[Released Packages](#) » | [Clipboard](#)

Copy or paste properties from one `UnityEngine.Material` to another. [More...](#)

## Static Public Member Functions

static void **Copy** (Material source)

Copy properties of specified material as source to paste later. [More...](#)

static bool **Paste** (Material target)

Paste properties of the source copied before to the specified target. [More...](#)

## Detailed Description

Copy or paste properties from one `UnityEngine.Material` to another.

Operate by `UnityEngine.Material` context menu "Copy Properties" and "Paste Properties". This works like the similar menu of `UnityEngine.Component`, and will change the shader. To keep valid after script reloaded by a temporary material as medium storage. Copy properties by built-in `Material.CopyPropertiesFromMaterial()` method.

## Member Function Documentation

static void **Copy** ( Material source )

static

Copy properties of specified material as source to paste later.

### Parameters

**source** Source.

static bool **Paste** ( Material target )

static

Paste properties of the source copied before to the specified target.

### Returns

false, if the source doesn't exist yet, otherwise true.

### Parameters

**target** Target.

Draw an auxiliary icon follows the cursor globally. [More...](#)

Inherits EditorWindow.

## Properties

static DragAndDropVisualMode **mode** [get, set]

The visual mode of cursor to show, to set this will override the icon. [More...](#)

static Texture **icon** [get, set]

The texture of cursor to show, to set this will override the mode. [More...](#)

static Vector2 **offset** [get, set]

The pixel offset between the auxiliary icon and the system cursor. [More...](#)

static Vector2 **size** [get, set]

The pixel size of the auxiliary icon, set Vector2.zero to use the default texture size. [More...](#)

## Detailed Description

Draw an auxiliary icon follows the cursor globally.

Used to provide a cross windows hint for reasons below:

1. DragAndDrop.visualMode only works when using UnityEditor.DragAndDrop system.
2. EditorGUIUtility.AddCursorRect() only applies inside the current window.

## Property Documentation

### DragAndDropVisualMode mode

static get set

The visual mode of cursor to show, to set this will override the icon.

The mode.

### Texture icon

static get set

The texture of cursor to show, to set this will override the mode.

The icon.

## Vector2 offset

static get set

The pixel offset between the auxiliary icon and the system cursor.

The offset.

## Vector2 size

static get set

The pixel size of the auxiliary icon, set `Vector2.zero` to use the default texture size.

The size.

# EditrixGUI

[Released Packages](#) » [3P MakerReleased Packages](#) » |

[BookmarkReleased Packages](#) » |

[ClipboardReleased Packages](#) » |

[Event DelegatorReleased Packages](#) » | [Notebook](#)

Include some convenient methods for editor GUI. [More...](#)

## Static Public Member Functions

static float	<b>GetLabelWidth</b> (float viewWidth=0f) Get the label width within the specified view width. <a href="#">More...</a>
static void	<b>MultiFloatField</b> (GUIContent label, GUIContent[] subLabels, float[] values, params GUILayoutOption[] options) Make a multi-control with text fields for entering multiple floats in the same line. <a href="#">More...</a>
static Vector4	<b>Vector4Field</b> (GUIContent label, Vector4 value, params GUILayoutOption[] options) Make an X, Y, Z and W field for entering a <code>UnityEngine.Vector4</code> . <a href="#">More...</a>
static Quaternion	<b>QuaternionField</b> (GUIContent label, Quaternion value, params GUILayoutOption[] options) Make an X, Y, Z and W field for entering a <code>UnityEngine.Quaternion</code> . <a href="#">More...</a>
static string	<b>SearchField</b> (string text, Action onPopup=null, params GUILayoutOption[] options) Make a toolbar style search field, optional to set a callback invoked when click the magnifier popup icon. <a href="#">More...</a>
static void	<b>ErrorField</b> (Rect position, GUIContent label, GUIContent error) Display an error message after the prefix label. <a href="#">More...</a>
static void	<b>SetIconForObject</b> (Object obj, Texture2D icon) Set the icon for the specified <code>UnityEngine.Object</code> to show in the Inspector or Project window. <a href="#">More...</a>
static Object[]	<b>CheckDragObjects</b> (Func< Object, bool > filter) Check if Drag'n'Drop performed with the filtered dragged <code>UnityEngine.Object</code> references. <a href="#">More...</a>
static Object[]	<b>CheckDragObjects</b> (Func< IEnumerable< Object >, IEnumerable< Object >> filter=null) Check if Drag'n'Drop performed with the filtered dragged <code>UnityEngine.Object</code> references. <a href="#">More...</a>
static string[]	<b>CheckDragPaths</b> (Func< string, bool > filter) Check if Drag'n'Drop performed with the filtered dragged string paths. <a href="#">More...</a>
static string[]	<b>CheckDragPaths</b> (Func< IEnumerable< string >, IEnumerable< string >> filter=null) Check if Drag'n'Drop performed with the filtered dragged string paths. <a href="#">More...</a>

## Detailed Description

Include some convenient methods for editor GUI.



## Member Function Documentation

---

**static float GetLabelWidth ( float viewWidth = 0f )**

static

Get the label width within the specified view width.

Commonly used for a prefix label in a resizable window. Use `EditorGUIUtility.currentViewWidth` instead if not assign.

### Returns

The label width.

### Parameters

**viewWidth** View width.

**static void MultiFloatField ( GUIContent label, GUIContent[] subLabels, float[] values, params GUILayoutOption[] options )**

static

Make a multi-control with text fields for entering multiple floats in the same line.

Wrap `EditorGUI.MultiFloatField()` as the GUI layout automatically.

### Parameters

**label** Main label.

**subLabels** Sub labels.

**values** Values.

**options** Layout options.

```
static Vector4 Vector4Field ( GUIContent label,  
                             Vector4 value,  
                             params GUILayoutOption[] options  
                             )
```

static

Make an X, Y, Z and W field for entering a `UnityEngine.Vector4`.

Copycat of `EditorGUILayout.Vector4Field()` but use **`MultiFloatField()`**. Contrast to original, this correct line wrap and indent level.

#### Returns

The new value.

#### Parameters

**label** Label.

**value** Value.

**options** Layout options.

```
static Quaternion QuaternionField ( GUIContent label,  
                                    Quaternion value,  
                                    params GUILayoutOption[] options  
                                    )
```

static

Make an X, Y, Z and W field for entering a `UnityEngine.Quaternion`.

Just looks like **`Vector4Field()`**.

#### Returns

The new value.

#### Parameters

**label** Label.

**value** Value.

**options** Layout options.

```
static string SearchField ( string          text,  
                           Action          onPopup = null,  
                           params GUILayoutOption[] options  
                           )
```

static

Make a toolbar style search field, optional to set a callback invoked when click the magnifier popup icon.

#### Returns

The new text.

#### Parameters

**text**        The text to edit.

**onPopup**    The popup callback.

**options**    Layout options.

```
static void ErrorField ( Rect      position,  
                        GUIContent label,  
                        GUIContent error  
                        )
```

static

Display an error message after the prefix label.

#### Parameters

**position**    Position.

**label**        Label.

**error**        Error message.

```
static void SetIconForObject ( Object      obj,
                               Texture2D icon
                               )
```

static

Set the icon for the specified `UnityEngine.Object` to show in the Inspector or Project window.

This wraps the internal `EditorGUIUtility.SetIconForObject()`:

1. It applies to all the same type objects, also the `UnityEditor.MonoScript` declares the type.
2. You may pass the script asset to set without any object instance.
3. To restore to default, pass a null icon.

The opposite methods below to get the icon:

1. `AssetPreview.GetMiniThumbnail()` to get by `UnityEngine.Object`.
2. `AssetPreview.GetMiniTypeThumbnail()` to get by `System.Type`.
3. `EditorGUIUtility.ObjectContent()` to get by both.

#### Parameters

**obj** Object.

**icon** Icon.

```
static Object [] CheckDragObjects ( Func< Object, bool > filter )
```

static

Check if Drag'n'Drop performed with the filtered dragged `UnityEngine.Object` references.

#### Returns

The filtered objects, only valid when drag performed, otherwise null.

#### Parameters

**filter** The callback invoked to filter each object only when drag updated.

```
static Object []
CheckDragObjects      ( Func< IEnumerable< Object >, IEnumerable< Object >> filter = null )
```

static

Check if Drag'n'Drop performed with the filtered dragged `UnityEngine.Object` references.

#### Returns

The filtered objects, only valid when drag performed, otherwise null.

#### Parameters

**filter** The callback invoked to filter all objects only when drag updated.

**static string [] CheckDragPaths ( Func< string, bool > filter )**

static

Check if Drag'n'Drop performed with the filtered dragged string paths.

**Returns**

The filtered paths, only valid when drag performed, otherwise null.

**Parameters**

**filter** The callback invoked to filter each path only when drag updated.

**static string []**

**CheckDragPaths** ( Func< IEnumerable< string >, IEnumerable< string >> filter = null )

static

Check if Drag'n'Drop performed with the filtered dragged string paths.

**Returns**

The filtered paths, only valid when drag performed, otherwise null.

**Parameters**

**filter** The callback invoked to filter all paths only when drag updated.

# EditrixStyle

[Released Packages](#) » [BookmarkReleased Packages](#) » |  
[ClipboardReleased Packages](#) » | [Notebook](#)

Extend editor GUI style, and include some premade icons or styles. [More...](#)

## Static Public Member Functions

static Texture2D	<b>LoadTexture</b> (int width, Color32[] pixels) Load a UnityEngine.Texture2D with color pixels for editor usage. <a href="#">More...</a>
static Texture2D	<b>LoadTexture</b> (string[] pixels, Dictionary< char, Color32 > colors, Color32 other) Trick to load a UnityEngine.Texture2D by parsing pixel string array. <a href="#">More...</a>
static Texture2D	<b>LoadIcon</b> (string[] pixels, byte pro=196, byte free=60) Trick to load icon UnityEngine.Texture2D by parsing pixel string array. <a href="#">More...</a>
static Texture2D	<b>LoadCursor</b> (string[] pixels) Trick to load cursor UnityEngine.Texture2D by parsing pixel string array. <a href="#">More...</a>

## Static Public Attributes

static readonly Texture2D	<b>splitterPixel</b> The splitter pixel texture with the same color as on the Inspector, 1 x 1. <a href="#">More...</a>
static readonly GUIStyle	<b>statusBar</b> The status bar style. <a href="#">More...</a>
static readonly Texture2D	<b>editIcon</b> The icon represents to edit, a pencil, 12 x 12. <a href="#">More...</a>
static readonly Texture2D	<b>copyIcon</b> The icon represents to copy, overlapping notes, 12 x 12. <a href="#">More...</a>
static readonly Texture2D	<b>removeIcon</b> The icon represents to remove, a moving out arrow, 12 x 12. <a href="#">More...</a>
static readonly Texture2D	<b>deleteIcon</b> The icon represents to delete, a trash can, 12 x 12. <a href="#">More...</a>
static readonly Texture2D	<b>aimIcon</b> The icon represents to aim, a front sight, 12 x 12. <a href="#">More...</a>
static readonly Texture2D	<b>hierarchyIcon</b> The icon represents a hierarchy, indent level lines, 12 x 12. <a href="#">More...</a>
static readonly Texture2D	<b>clipboardIcon</b> The icon represents a clipboard, 12 x 12. <a href="#">More...</a>
static readonly Texture2D	<b>bookmarkIcon</b> The icon represents a bookmark, 12 x 12. <a href="#">More...</a>
static readonly Texture2D	<b>questionIcon</b> The icon represents the Super Mario question mark, 12 x 12. <a href="#">More...</a>

static readonly Texture2D	<b>linkIcon</b>	The icon represents to link, a connected chain, 9 x 9. <a href="#">More...</a>
static readonly Texture2D	<b>unlinkIcon</b>	The icon represents to unlink, a broken chain, 9 x 9. <a href="#">More...</a>
static readonly Texture2D	<b>stopCursor</b>	The aux cursor represents a stop mark, 20 x 20. <a href="#">More...</a>
static readonly Texture2D	<b>copyCursor</b>	The aux cursor with a plus symbol, 19 x 15. <a href="#">More...</a>
static readonly Texture2D	<b>linkCursor</b>	The aux cursor with an arrow, 19 x 15. <a href="#">More...</a>
static readonly Texture2D	<b>moveCursor</b>	The aux cursor represents a dotted frame, 13 x 9. <a href="#">More...</a>

## Detailed Description

Extend editor GUI style, and include some premade icons or styles.

## Member Function Documentation

**static Texture2D LoadTexture ( int width, Color32[] pixels )**
static

Load a `UnityEngine.Texture2D` with color `pixels` for editor usage.

Return the texture with the same `pixels` created by this if exists, otherwise create new one.

**Returns**

The texture.

**Parameters**

**width** Width.

**pixels** Pixels.

```
static Texture2D LoadTexture ( string[] pixels,
                             Dictionary< char, Color32 > colors,
                             Color32 other
                             )
```

static

Trick to load a UnityEngine.Texture2D by parsing pixel string array.

Array length as texture height, element string length as width. Set each char pixel by colors map if existing, otherwise the other color. Return the texture with the same pixels created by this if exists, otherwise create new one.

### Returns

The texture.

### Parameters

**pixels** Pixel bits.

**colors** Colors map.

**other** Other.

```
static Texture2D LoadIcon ( string[] pixels,
                            byte pro = 196,
                            byte free = 60
                            )
```

static

Trick to load icon UnityEngine.Texture2D by parsing pixel string array.

Array length as icon height, element string length as width. Any space char as transparent, others color light if pro skin, otherwise dark. Return the texture with the same pixels created by this if exists, otherwise create new one.

### Returns

The icon UnityEngine.Texture2D.

### Parameters

**pixels** Pixel bits.

**pro** Grayscale for pro skin.

**free** Grayscale for free skin.



### static Texture2D LoadCursor ( string[] pixels )

static

Trick to load cursor `UnityEngine.Texture2D` by parsing pixel string array.

Array length as cursor height, element string length as width. Any '1' char as white, '0' as black, otherwise transparent. Return the texture with the same pixels created by this if exists, otherwise create new one.

#### Returns

The cursor `UnityEngine.Texture2D`.

#### Parameters

**pixels** Pixel bits.

## Member Data Documentation

### readonly Texture2D splitterPixel

static

The splitter pixel texture with the same color as on the Inspector, 1 x 1.

### readonly GUIStyle statusBar

static

The status bar style.

Text colored light if pro skin, otherwise dark, with darken background anyway.

### readonly Texture2D editIcon

static

The icon represents to edit, a pencil, 12 x 12.

### readonly Texture2D copyIcon

static

The icon represents to copy, overlapping notes, 12 x 12.

### readonly Texture2D removeIcon

static

The icon represents to remove, a moving out arrow, 12 x 12.

**readonly Texture2D deletelcon**

static

The icon represents to delete, a trash can, 12 x 12.

**readonly Texture2D aimlcon**

static

The icon represents to aim, a front sight, 12 x 12.

**readonly Texture2D hierarchyIcon**

static

The icon represents a hierarchy, indent level lines, 12 x 12.

**readonly Texture2D clipboardIcon**

static

The icon represents a clipboard, 12 x 12.

**readonly Texture2D bookmarkIcon**

static

The icon represents a bookmark, 12 x 12.

**readonly Texture2D questionIcon**

static

The icon represents the Super Mario question mark, 12 x 12.

**readonly Texture2D linkIcon**

static

The icon represents to link, a connected chain, 9 x 9.

**readonly Texture2D unlinkIcon**

static

The icon represents to unlink, a broken chain, 9 x 9.

**readonly Texture2D stopCursor**

static

The aux cursor represents a stop mark, 20 x 20.

**readonly Texture2D copyCursor**

static

The aux cursor with a plus symbol, 19 x 15.

**readonly Texture2D linkCursor**

static

The aux cursor with an arrow, 19 x 15.

**readonly Texture2D moveCursor**

static

The aux cursor represents a dotted frame, 13 x 9.

# MenuExtension

Released Packages » 3P MakerReleased Packages » |  
BookmarkReleased Packages » |  
ClipboardReleased Packages » |  
Editor AiderReleased Packages » |  
Event DelegatorSpecific Classes » |  
ExtensionsReleased Packages » | Notebook

**Extension** methods for UnityEditor.GenericMenu. [More...](#)

## Static Public Member Functions

static void	<b>AddItem&lt; T &gt;</b> (this GenericMenu menu, string content, Action< T > func, T param, bool check=false, bool enable=true) Add an item to the menu, it will be disabled if the function isn't assigned. <a href="#">More...</a>
static void	<b>AddItem</b> (this GenericMenu menu, string content, Action func=null, bool check=false, bool enable=true) Add an item to the menu, it will be disabled if the function isn't assigned. <a href="#">More...</a>

## Detailed Description

**Extension** methods for UnityEditor.GenericMenu.

3

## Member Function Documentation

```
static void AddItem< T > ( this GenericMenu menu,
                        string          content,
                        Action< T >     func,
                        T               param,
                        bool            check = false,
                        bool            enable = true
                        )
```

static

Add an item to the menu, it will be disabled if the function isn't assigned.

#### Parameters

**menu** Menu.

**content** Content.

**func** The function to call when the menu item is selected.

**param** The parameter to pass to the function.

**check** Whether to show the item is currently activated.

**enable** If set to false to enforce disabled.

#### Template Parameters

**T** The parameter type.

```
static void AddItem ( this GenericMenu menu,
                    string          content,
                    Action          func = null,
                    bool            check = false,
                    bool            enable = true
                    )
```

static

Add an item to the menu, it will be disabled if the function isn't assigned.

#### Parameters

**menu** Menu.

**content** Content.

**func** The function to call when the menu item is selected.

**check** Whether to show the item is currently activated.

**enable** If set to false to enforce disabled.

# ComponentExtension

[Specific Classes](#) » [ExtensionsReleased Packages](#) » |  
[Clipboard](#)

**Extension** methods for `UnityEngine.Component`. [More...](#)

## Static Public Member Functions

static string **GetPath** (this Transform transform, string separator="/")  
Get the hierarchy path, used for `log` or `GameObject.Find()` with the default separator.  
[More...](#)

static Transform[] **GetParents** (this Transform transform)  
Get the hierarchy transforms, sorted from self to the root. [More...](#)

static Quaternion **TransformQuat** (this Transform transform, Quaternion rotation)  
Transforms rotation from local space to world space. [More...](#)

static Quaternion **InverseTransformQuat** (this Transform transform, Quaternion rotation)  
Transforms position from world space to local space, opposite of **TransformQuat()**.  
[More...](#)

static void **SetSorting** (this Component component, int order)  
Set the sorting order of the UI element inside a `UnityEngine.Canvas`. [More...](#)

## Detailed Description

**Extension** methods for `UnityEngine.Component`.

## Member Function Documentation

```
static string GetPath ( this Transform transform,
                        string separator = "/"
                        )
```

static

Get the hierarchy path, used for `log` or `GameObject.Find()` with the default separator.

### Returns

The hierarchy path.

### Parameters

**transform** Transform.

**separator** Separator.

**static Transform [] GetParents ( this Transform transform )**

static

Get the hierarchy transforms, sorted from self to the root.

**Returns**

The hierarchy transforms.

**Parameters**

**transform** Transform.

**static Quaternion TransformQuat ( this Transform transform,  
Quaternion rotation  
)**

static

Transforms rotation from local space to world space.

**Returns**

The world rotation.

**Parameters**

**transform** Transform.

**rotation** Local rotation.

**static Quaternion InverseTransformQuat ( this Transform transform,  
Quaternion rotation  
)**

static

Transforms position from world space to local space, opposite of **TransformQuat()**.

**Returns**

The local rotation.

**Parameters**

**transform** Transform.

**rotation** World rotation.

```
static void SetSorting ( this Component component,  
                        int order  
                        )
```

static

Set the sorting order of the UI element inside a `UnityEngine.Canvas`.

This only apply the sub canvas in a root canvas. Check to add it with `UnityEngine.UI.GraphicRaycaster` if not existing.

#### Parameters

**component** Component.

**order** Sorting order.



# TypeExtension

[Released Packages](#) » [3P MakerReleased Packages](#) » |

[BookmarkReleased Packages](#) » |

[ClipboardReleased Packages](#) » |

[Editor AiderReleased Packages](#) » |

[Event DelegatorSpecific Classes](#) » |

[ExtensionsReleased Packages](#) » |

[NotebookReleased Packages](#) » | [Www Loader](#)

**Extension** methods for `System.Type`. [More...](#)

## Static Public Member Functions

static string **GetPrettyName** (this Type type, bool full=false)  
Get a pretty readable name of the type, even generic, optional to use the full name.  
[More...](#)

static Type **GetNestedType** (this Type type, string name, bool isStatic)  
Get the named public or nonpublic nested type of the specified type. [More...](#)

static FieldInfo **GetField** (this Type type, string name, bool isStatic, Type fieldType=null)  
Get the named public or nonpublic FieldInfo of the specified type. [More...](#)

static PropertyInfo **GetProperty** (this Type type, string name, bool isStatic, Type propertyType=null, params Type[] indexTypes)  
Get the named public or nonpublic PropertyInfo of the specified type. [More...](#)

static MethodInfo **GetMethod** (this Type type, string name, bool isStatic, Type returnType=null, params Type[] paramTypes)  
Get the named public or nonpublic MethodInfo of the specified type. [More...](#)

static Type[] **GetParents** (this Type type)  
Get the parent hierarchy array, sorted from self to root type. [More...](#)

static Type[] **GetChildren** (this Type type, bool deep=false)  
Get all child types, excluding self, optional to find deep or directly inheritance only. [More...](#)

static Type **GetItemType** (this Type type)  
Return the element type of an array or list type, otherwise null. [More...](#)

static object **GetDefault** (this Type type)  
Get the default value of the type, just like `default(T)`. [More...](#)

static bool **IsCreatable** (this Type type, bool exception=false)  
Determine if able to create an instance of the type. [More...](#)

## Detailed Description

**Extension** methods for `System.Type`.

## Member Function Documentation

---

```
static string GetPrettyName ( this Type type,
                             bool      full = false
                             )
```

static

Get a pretty readable name of the type, even generic, optional to use the full name.

This doesn't handle anonymous types.

### Returns

The pretty name.

### Parameters

**type** Type.

**full** If set to true use the full name.

```
static Type GetNestedType ( this Type type,
                            string    name,
                            bool      isStatic
                            )
```

static

Get the named public or nonpublic nested type of the specified type.

### Returns

The nested type.

### Parameters

**type** Type.

**name** Type name.

**isStatic** If to get a static type.

```
static FieldInfo GetField ( this Type type,
                           string name,
                           bool isStatic,
                           Type fieldType = null
                           )
```

static

Get the named public or nonpublic FieldInfo of the specified type.

#### Returns

The field.

#### Parameters

**type** Type.

**name** Field name.

**isStatic** If to get a static field.

**fieldType** Field type.

```
static PropertyInfo GetProperty ( this Type type,
                                   string name,
                                   bool isStatic,
                                   Type propertyType = null,
                                   params Type[] indexTypes
                                   )
```

static

Get the named public or nonpublic PropertyInfo of the specified type.

#### Returns

The property.

#### Parameters

**type** Type.

**name** Property name.

**isStatic** If to get a static property.

**propertyType** Property type.

**indexTypes** Index types.

```
static MethodInfo GetMethod ( this Type    type,
                             string      name,
                             bool        isStatic,
                             Type        returnType = null,
                             params Type[] paramTypes
                             )
```

static

Get the named public or nonpublic MethodInfo of the specified type.

#### Returns

The method.

#### Parameters

**type** Type.  
**name** Method name.  
**isStatic** If to get a static method.  
**returnType** Return type.  
**paramTypes** Parameter types.

```
static Type [] GetParents ( this Type type )
```

static

Get the parent hierarchy array, sorted from self to root type.

#### Returns

The parent hierarchy array.

#### Parameters

**type** Type.

```
static Type [] GetChildren ( this Type type,
                             bool        deep = false
                             )
```

static

Get all child types, excluding self, optional to find deep or directly inheritance only.

#### Returns

The child types.

#### Parameters

**type** Type.  
**deep** If set to true deep.

**static Type GetItemType ( this Type type )**

static

Return the element type of an array or list type, otherwise null.

## Returns

The element type.

## Parameters

**type** Type.

**static object** **GetDefault** ( **this Type** **type** )

static

Get the default value of the type, just like `default(T)`.

## Returns

The default value.

## Parameters

**type** Type.

```
static bool IsCreatable ( this Type type,
                        bool exception = false
                      )
```

static

Determine if able to create an instance of the type.

Optional to throw an exception message or just return `false` if invalid. This only checks some basic conditions and might be not precise.

The current conditions below:

1. Return `false` only if it's interface, abstract, generic definition, delegate.
2. Recurse to check the element type of an array type.
3. Recurse to check the generic arguments of a list or dictionary type.

## Returns

true, if creatable, false otherwise.

## Parameters

**type**      Type.

**exception** Flag to throw an exception or return false.