

# TAB MANAGER CODE DOCUMENTATION

Code Documentation for the Unity3D plugin built by Chimeforest.

Updated for TabManager 1.2.0

## CONTENTS

- Introduction ..... 2
  - TabContainer Hierarchy ..... 2
  - Some DOs and DO NOTs..... 2
- 1 ComponentExtention ..... 3
  - 1.1 Functions ..... 3
- 2 Helper..... 4
  - 2.1 Functions ..... 4
- 3 Tab ..... 5
  - 3.1 Variables ..... 5
  - 3.2 Functions ..... 5
- 4 TabButton ..... 7
  - 4.1 Functions ..... 7
- 5 TabConatiner ..... 8
  - 5.1 Functions ..... 8
- 6 TabConfig .....10
  - 6.1 Constants .....10
  - 6.2 Variables.....11
- 7 TabContext.....12
  - 7.1 Constants .....12
  - 7.2 Functions .....12
- 8 TabInspectors.....13
  - 8.1 Classes .....13
- 9 TabManager .....14
  - 9.1 Constants .....14
  - 9.2 Variables.....14
  - 9.3 Functions .....16
- 10 TabUtilMenu.....23
  - 10.1 Variables.....23

## INTRODUCTION

Hello and thank you for your purchase of this plugin. In this document I've summarized all the classes and functions which you can use to get the most out of this plugin, as well as some information on how and when to use them.

So without further ado, let's get started.

First thing that you will need to know is that all the classes are part of the "Chimeforest.TabManager" namespace, so you'll want to make sure that you are using that in every script where you manipulate tabs.

Second, here are the classes of most interest:

- TabManager: Use this class to add, move, and remove tabs in a tab container.
- TabContext: In here is the code to create a Tab Container.
- TabButton: You can look at this for an example of how to add functionality to buttons.
- Tab: Has the Tab index and TabManager for easy reference and direct manipulation of tabs.

Third, the Tab Container has a semi rigid hierarchy (more info below). If you plan on creating your own Tab Containers from scratch, or feel like editing an existing one, you should become familiar with the hierarchy.

## TABCONTAINER HIERARCHY

For the most part, the Tab Container Hierarchy is fairly flexible; as long as the objects reference each other correctly many of them can be placed anywhere you want them in your project. However, there are a couple of rules which you must follow:

- Only put buttons for tabs under a TabButtons object.
- Only put panels for tabs under a TabPanels object.
- Tab buttons MUST go under a TabButton object.
- Tab panels MUST go under a TabPanels object.

Other than that, you can do pretty much whatever you want.

## SOME DOS AND DO NOTS

### DO

- 'using Chimeforest.TabManager'
- Use your own graphics.
- Write your own code.
- Look at the code I wrote to see how I made this plugin.
- Look at the examples and try a few out.
- Make suggestions on how to improve the plugin.
- Tell me about bugs, so I can fix them asap.
- Send me a message if you have questions: [chimeforestproductions@gmail.com](mailto:chimeforestproductions@gmail.com)

### DO NOT

- Use a function or class which you haven't read about.
- Detach tab scripts from objects.
- Make modifications of this plugin and try to sell it as your own work.
- Blame me when something goes wrong; instead, post the question to <http://answers.unity3d.com/> and tag me in the question (@chimeforest) and I'll get to it when I can =]
- Use this plugin for world domination.

## 1 COMPONENTEXTENTION

Public class in Chimeforest.TabManager

This class extends the functionality of components with a couple functions copy and replace components on an object.

### 1.1 FUNCTIONS

|                 |        |      |  |
|-----------------|--------|------|--|
| AddComponent<T> | public | void | Adds a copy of another component to object.            |
| GetCopyOf<T>    | public | void | Returns a copy of the component.                       |
| Replace<T>      | public | void | Replaces an existing component with a copy of another. |

#### 1.1.1 ADDCOMPONENT<T>

public static T **AddComponent**<T>(this GameObject go, T toAdd) where T : Component

##### 1.1.1.1 PARAMETERS

|       |            |   |
|-------|------------|---|
| go    | GameObject | The GameObject which has the component to copy. |
| toAad | T          | The component type you want to copy             |

Uses GetCopyOf<T> to copy a component onto the object.

#### 1.1.2 GETCOPYOF<T>

public static T **GetCopyOf**<T>(this Component comp, T other) where T : Component

##### 1.1.2.1 PARAMETERS

|       |           |   |
|-------|-----------|---|
| comp  | Component | The component to copy.                  |
| other | T         | The type of component you want to copy. |

This function makes a copy of a component and returns it.

#### 1.1.3 REPLACECOMPONENT<T>

public static T **ReplaceComponent**<T>(this GameObject go, T toReplace) where T : Component

##### 1.1.3.1 PARAMETERS

|           |            |   |
|-----------|------------|---|
| go        | GameObject | The GameObject which has the component to copy. |
| toReplace | T          | The component type you want to copy.            |

This function replaces a component on one object with a copy of the same component from another object. If the original object does not have the specified component, it still be copied.

## 2 HELPER

Public class in `Chimeforest.TabManager`

This class a couple helper functions to assist with calculations and gathering info.

### 2.1 FUNCTIONS

|  |        |                                     |  |
|--|--------|-------------------------------------|--|
| <code>CenterAndStretchRectTransform</code> | public | void                                | Stretches and centers a <code>RectTransform</code> on its parent.                |
| <code>GetChildren</code>                   | public | <code>List&lt;GameObject&gt;</code> | Gets all the children <code>GameObjects</code> and returns a <code>List</code> . |

#### 2.1.1 CENTERANDSTRETCHRECTTRANSFORM

public static void **CenterAndStrecthRectTransform**(GameObject gameobject)

public static void **CenterAndStrecthRectTransform**(RectTransform rectTransf)

##### 2.1.1.1 PARAMETERS

|                         |                            |  |
|-------------------------|----------------------------|--|
| <code>gameobject</code> | <code>GameObject</code>    | The <code>GameObject</code> with the <code>RectTransform</code> to center and stretch. |
| <code>rectTransf</code> | <code>RectTransform</code> | The <code>RectTransform</code> to center and stretch.                                  |

This function Stretches and centers a `RectTransform` on its parent object. There are two overrides for this function, one accepts a `RectTransform`, the other a `GameObject` with a `RectTransform`.

#### 2.1.2 GETCHILDREN

public static `List<GameObject>` **GetChildren**(GameObject gameObj)

##### 2.1.2.1 PARAMETERS

|                      |                         |   |
|----------------------|-------------------------|---|
| <code>gameObj</code> | <code>GameObject</code> | The <code>GameObject</code> to get children from. |
|----------------------|-------------------------|---|

This function gets all the children game objects which are attached to a `GameObject`.

## 3 TAB

Public class in `Chimeforest.TabManager` inherits from `MonoBehavior`

This class is used as a hook for the Tab inspector menu. It also contains a couple useful things for programming with tabs. It is attached to every tab panel and tab button.

### 3.1 VARIABLES

|     |            |     |                        |
|-----|------------|-----|------------------------|
| tab | serialized | int | The index of this tab. |
|-----|------------|-----|------------------------|

#### 3.1.1 TAB

```
[SerializeField] int tab;
```

This variable stores the tab index on the tab for quick reference. It can't be used directly - instead use the `GetTab` and `SetTab` functions for that.

### 3.2 FUNCTIONS

|                         |        |                         |   |
|-------------------------|--------|-------------------------|---|
| <code>GetTab</code>     | public | int                     | Gets the tab variable for this tab.               |
| <code>IsSelected</code> | public | bool                    | Returns whether the tab is the current tab.       |
| <code>SetTab</code>     | public | void                    | Sets the tab variable for this tab.               |
| <code>TabManager</code> | public | <code>TabManager</code> | Returns the <code>TabManager</code> for this tab. |

#### 3.2.1 GETTAB

```
public int GetTab();
```

Gets the tab variable for this tab.

#### 3.2.2 SETTAB

```
public void SetTab(int tabIndex);
```

##### 3.2.2.1 PARAMETERS

|          |     |                                   |
|----------|-----|-----------------------------------|
| tabIndex | int | The tab index to set this tab to. |
|----------|-----|-----------------------------------|

Sets the tab variable for this tab. You should only use this function if you know what you are doing. Improper use of this function can mess up the tab structure.

#### 3.2.3 ISSELECTED

```
public bool IsSelected();
```

Checks with the `TabManager` to see if the tab is currently selected as the active tab.

---

### 3.2.4 TABMANAGER

```
public TabManager TabManager();
```

Returns a reference to the TabManager which controls this tab. It does this by asking its parent (a TabUtilMenu) which TabManager it uses.

## 4 TABBUTTON

Public class in `Chimeforest.TabManager` inherits from `MonoBehavior`

This class is attached to tab buttons. It tells the button to switch tabs when clicked.

If you want something to happen when a button is clicked, the code for this class is a great template; just do the following:

1. Make a copy of `TabButton.cs`.
2. Give it its own name.
3. Use the `Chimeforest.TabManager` namespace (using `Chimeforest.TabManager;`).
4. Change the `OnClick` function to suit your needs.
5. Attach it to the tab buttons you want it to affect.

### 4.1 FUNCTIONS

|         |         |      |   |
|---------|---------|------|---|
| Awake   | private | void | When the component wakes up, add a listener.                          |
| OnClick | private | void | When clicked, tell the <code>TabManager</code> to switch to this tab. |

## 5 TABCONATINER

Public class in Chimeforest.TabManager inherits from UnityEditor.Editor

This class contains the code to make a TabConatiner.

### 5.1 FUNCTIONS

|                                     |        |            |  |
|-------------------------------------|--------|------------|--|
| CreateBLANKTabContainer             | public | GameObject | Returns a TabContainer with no Tabs.                   |
| CreateTabContainer                  | public | GameObject | Returns a TabConainter with 1 Tab.                     |
| CreateTabContainerWithObjects       | public | GameObject | Returns a TabConainter with prefab buttons and panels. |
| CreateTabContainerWithResourcePaths | public | GameObject | Returns a TabConainter with prefab buttons and panels. |

#### 5.1.1 CREATEBLANKTABCONTAINER

```
public static GameObject CreateBLANKTabContainer();
```

Returns a completely made Tab Container with NO Tabs. Can be used during runtime to create a Tab Container on the fly.

#### 5.1.2 CREATETABCONTAINER

```
public static GameObject CreateTabContainer();
```

Returns a completely made Tab Container with one Tab. Used by AddTabContainerToMenu in the TabContext class. Can be used during runtime to create a Tab Container on the fly.

#### 5.1.3 CREATETABCONTAINERWITHOBJECTS

```
public static GameObject CreateTabContainerWithObjects();
```

##### 5.1.3.1 PARAMETERS

|               |            |  |
|---------------|------------|--|
| prefab_Button | GameObject | The GameObject to use as a prefab for the TabButtons |
| prefab_Panel  | GameObject | The GameObject to use as a prefab for the TabPanels  |

Returns a completely made Tab Container with one Tab. Can be used during runtime to create a Tab Container on the fly. Use your own GameObjects for the buttons and/or Panels. Make sure the appropriate components are attached to them. You can use the default ones in Plugins/Chimeforest/TabManager/Resources as a template.

---

#### 5.1.4 CREATETABCONTAINERWITHRESOUCEPATHS

```
public static GameObject CreateTabContainerWithResourcePaths();
```

---

##### 5.1.4.1 PARAMETERS

|               |            |  |
|---------------|------------|--|
| prefab_Button | GameObject | The GameObject to use as a prefab for the TabButtons |
| prefab_Panel  | GameObject | The GameObject to use as a prefab for the TabPanels  |

Returns a completely made Tab Container with one Tab. Can be used during runtime to create a Tab Container on the fly. Reference your own prefabs in a Resources folder for the buttons and/or Panels. Make sure the appropriate components are attached to them. You can use the default ones in Plugins/Chimeforest/TabManager/Resources as a template.

## 6 TABCONFIG

class in `Chimeforest.TabManager`

This class holds the configurations/settings for this plugin.

### 6.1 CONSTANTS

|   |        |      |   |
|---|--------|------|---|
| <code>context_ContainerStretchFill</code> | public | bool | A setting for whether or not to resize new tab containers               |
| <code>manager_ForceRename</code>          | public | bool | Adds the option to create Tab Containers from the context menu.         |
| <code>manager_SwitchToNewTab</code>       | public | bool | Adds the option to add Tabs to existing Containers to the Context Menu. |

---

#### 6.1.1 CONTAINER\_STRETCHFILL

```
public const bool context_ContainerStretchFill
```

This is a setting for the `TabContext` class. The default value is true. If it is set to true, new Tab Containers which are made will stretch to fill whatever parent object they are attached to. If it is false, then new Tab Containers will be the default size of 100x100pixels.

---

#### 6.1.2 FORCE\_RENAME

```
public const bool manager_ForceRename
```

This is a setting for the `TabManager` class. The default is true. If this constant is true, every time the `ReNumberTabs` function is called, it will also rename the tab buttons and panels to match. This is to keep everything organized. If it is false, they will not be renamed, and the user/programmer can name them whatever they wish.

---

#### 6.1.3 SWITCH\_TO\_NEW\_TAB

```
public const bool manager_SwitchToNewTab
```

This is a setting for the `TabManager` class. The default is false. If this constant is true, in the editor when a new tab is made it will become active and selected. If not in the editor, then the new tab will become active, but not selected. If it is false, then the active tab will remain the same.

## 6.2 VARIABLES

|                                   |                     |                     |  |
|-----------------------------------|---------------------|---------------------|--|
| <code>prefabPath_Button</code>    | <code>public</code> | <code>string</code> | Stores the location of the button prefab.    |
| <code>prefabPath_Container</code> | <code>public</code> | <code>string</code> | Stores the location of the container prefab. |
| <code>prefabPath_Panel</code>     | <code>public</code> | <code>string</code> | Stores the location of the panel prefab.     |

---

### 6.2.1 PREFABPATH\_BUTTON

`public bool manager_SwitchToNewTab`

This points to the button prefab used by this plugin. Change this if you want to use your own prefab.

---

### 6.2.2 PREFABPATH\_CONATINER

`public bool manager_SwitchToNewTab`

This points to the TabContainer prefab used by this plugin. It is not recommended to change this.

---

### 6.2.3 PREFABPATH\_PANEL

`public bool manager_SwitchToNewTab`

This points to the panel prefab used by this plugin. Change this if you want to use your own prefab.

## 7 TABCONTEXT

Public class in `Chimeforest.TabManager` inherits from `UnityEditor.Editor`

This class lets the user create Tabs and Tab Containers from the context menu in the Hierarchy window, or from the GameObject menu.

The only reason to look at this class is to see how I added something to the context menu.

### 7.1 CONSTANTS

|                                    |                      |                   |   |
|------------------------------------|----------------------|-------------------|---|
| <code>CONTAINER_STRETCHFILL</code> | <code>private</code> | <code>bool</code> | A setting for whether or not to resize new tab containers |
|------------------------------------|----------------------|-------------------|---|

Constants are used as settings for this plugin; to change them go to the `TabConfig` class.

### 7.2 FUNCTIONS

|                                    |                      |                   |   |
|------------------------------------|----------------------|-------------------|---|
| <code>AddTabContainerToMenu</code> | <code>private</code> | <code>void</code> | Adds the option to create Tab Containers from the context menu.         |
| <code>AddTabToMenu</code>          | <code>private</code> | <code>void</code> | Adds the option to add Tabs to existing Containers to the Context Menu. |

## 8 TABINSPECTORS

Public classes in `Chimeforest.TabManager` inherits from `UnityEditor.Editor`

These classes add GUIs to the inspectors for Tabs and TabUtilMenus. They also automatically fix some bugs with undoing making, removing, and moving Tabs. You shouldn't need to mess with anything in this class.

### 8.1 CLASSES

|                               |                     |   |
|-------------------------------|---------------------|---|
| <code>TabUtilInspector</code> | <code>public</code> | Adds GUI to TabUtilMenu and auto-fixes undo bugs. |
| <code>TabInspector</code>     | <code>public</code> | Adds GUI to Tab and auto-fixes undo bugs.         |

## 9 TABMANAGER

Public class in `Chimeforest.TabManager` inherits from `MonoBehavior`

This class is where all the action happens. The `TabManager` is the base of operations; it keeps the tabs orderly, adds new tabs, and removes old tabs. Each Tab Container has one. If you plan to program anything besides the most basic functions with this plugin, you should become very familiar with how this class operates. Do not use any of the functions or set any variables until you understand them.

### 9.1 CONSTANTS

|                                |         |      |   |
|--------------------------------|---------|------|---|
| <code>FORCE_RENAME</code>      | private | bool | A setting, if true, every time <code>ReNumberTabs</code> is called, it will also rename the tab buttons and panels. |
| <code>SWITCH_TO_NEW_TAB</code> | private | bool | A setting, if true, when a new tab is made it will become active and selected.                                      |

Constants are used as settings for this plugin; to change them go to the `TabConfig` class.

### 9.2 VARIABLES

|                           |        |                                     |   |
|---------------------------|--------|-------------------------------------|---|
| <code>buttons</code>      | public | <code>List&lt;GameObject&gt;</code> | Keeps track of the children of <code>TabButtons</code> .                      |
| <code>currentTab</code>   | public | int                                 | Stores the index of the currently active Tab.                                 |
| <code>panels</code>       | public | <code>List&lt;GameObject&gt;</code> | Keeps track of the children of <code>TabPanels</code> .                       |
| <code>prefabButton</code> | public | <code>GameObject</code>             | <i>Optional.</i> When adding a new tab it will use this object as the button. |
| <code>prefabPanel</code>  | public | <code>GameObject</code>             | <i>Optional.</i> When adding a new tab it will use this object as the panel.  |
| <code>TabButtons</code>   | public | <code>GameObject</code>             | References the <code>GameObject</code> which holds all of the tab buttons.    |
| <code>TabPanels</code>    | public | <code>GameObject</code>             | References the <code>GameObject</code> which holds all of the tabs panels.    |

#### 9.2.1 BUTTONS

Public `List<GameObject> buttons`

This variable keeps track of all the buttons in this tab container in a `List`. The list is updated whenever the `SetLists` or `ReNumberTabs` function is called, which is fairly often.

#### 9.2.2 CURRENTTAB

Public int `currentTab`

This variable stores the currently active tab. Do not set this variable, instead use the `SwitchToTab` function.

---

### 9.2.3 PANELS

Public List<GameObject> panels

This variable keeps track of all the panels in this tab container in a List. The list is updated whenever the SetLists or ReNumberTabs function is called, which is fairly often.

---

### 9.2.4 PREFABBUTTON

Public GameObject prefabButton

When making a new tab it will try to use this object as a button. If this object is null it will use the default tab button. If you create your own button, make sure the appropriate components are attached to it. You can use the default one in Plugins/Chimeforest/TabManager/Resources as a template.

---

### 9.2.5 PREFABPANEL

Public GameObject prefabPanel

When making a new tab it will try to use this object as a panel. If this object is null it will use the default tab panel. If you create your own panel, make sure the appropriate components are attached to it. You can use the default one in Plugins/Chimeforest/TabManager/Resources as a template.

---

### 9.2.6 TABBUTTONS

Public GameObject TabButtons

This variable references the object which contains all the tab buttons for this tab container. Only buttons should be the children of the object, and it must have a TabUtilMenu script attached to it.

---

### 9.2.7 TABPANELS

Public GameObject TabPanels

This variable references the object which contains all the tab panels for this tab container. Only tab panels should be the children of the object, and it must have a TabUtilMenu script attached to it.

### 9.3 FUNCTIONS

|                              |         |      |  |
|------------------------------|---------|------|--|
| AddTabBegin                  | public  | void | Adds a new tab to the beginning.                                     |
| AddTabBeginWithObjects       | public  | void | Adds a new tab to the beginning.                                     |
| AddTabBeginWithResourcePaths | public  | void | Adds a new tab to the beginning.                                     |
| AddTabEnd                    | public  | void | Adds a new tab at the end.   |
| AddTabEndWithObjects         | public  | void | Adds a new tab at the end.   |
| AddTabEndWithResourcePaths   | public  | void | Adds a new tab at the end.   |
| CountActivePanels            | public  | int  | Returns the number of active tab panels.                             |
| InsertTab                    | public  | void | Inserts a tab at the specified index.                                |
| InsertTabWithObjects         | public  | void | Inserts a tab at the specified index.                                |
| InsertTabWithResourcePaths   | public  | void | Inserts a tab at the specified index.                                |
| MoveTabDown                  | public  | void | Moves a tab down in the index.                                       |
| MoveTabUp                    | public  | void | Moves a tab up in the index.   |
| RemoveAllTabs                | public  | void | Removes all of the tabs.   |
| RemoveTab                    | public  | void | Removes a tab at the specified index.                                |
| RemoveTabBegin               | public  | void | Removes the first tab.   |
| RemoveTabEnd                 | public  | void | Removes the last tab.  |
| ReNumberTabs                 | public  | void | Renumbers and renames all the tabs in this container.                |
| ReskinAllButtons             | public  | void | Changes the image and text style of all tab buttons.                 |
| ReskinAllPanels              | public  | void | Changes the image all tab panels.                                    |
| ReskinAllTabs                | public  | void | Changes the image and text style of all tabs.                        |
| ReskinButton                 | public  | void | Changes a tab button's image and text style.                         |
| ReskinPanel                  | public  | void | Changes a tab panel's image.   |
| ReskinTab                    | public  | void | Changes a tab's image and text style.                                |
| SetLists                     | public  | void | Sets the panels and buttons variables                                |
| SetTabsIndexViaPanels        | public  | void | Forces the siblingIndex to match Tab.tab.                            |
| Start                        | private | void | Prepares the tab container to be used during runtime.                |
| SwitchToTab                  | public  | void | Makes the tab at the specified index the active tab.                 |
| TabPanelIndexesMatch         | public  | bool | Returned whether the siblingIndex of the tabs matches their Tab.tab. |

---

### 9.3.1 ADDTABBEGIN

```
public void AddTabBegin(string btnTxt);
```

#### 9.3.1.1 PARAMETERS

|        |        |   |
|--------|--------|---|
| btnTxt | string | <i>Optional.</i> Sets the text for the button of the new tab. Default is “TabButton”. |
|--------|--------|---|

This function uses the InsertTab function to add a tab to the beginning of the tab list.

---

### 9.3.2 ADDTABBEGINWITHOBJECTS

```
public void AddTabBeginObjects(string btnTxt, GameObject resource_ButtonPrefab , GameObject resource_PanelPrefab);
```

#### 9.3.2.1 PARAMETERS

|                       |            |   |
|-----------------------|------------|---|
| btnTxt                | string     | <i>Optional.</i> Sets the text for the button of the new tab. Default is “TabButton”. |
| resource_ButtonPrefab | GameObject | <i>Optional.</i> Uses this object for the new tab’s button.                           |
| resource_PanelPrefab  | GameObject | <i>Optional.</i> Uses this object for the new tab’s panel.                            |

This function uses the InsertTab function to add a tab to the beginning of the tab list. It takes two optional GameObjects which is uses for the new tab’s button and panel.

---

### 9.3.3 ADDTABBEGINWITHRESOURCEPATHS

```
public void AddTabBeginWithResourcePaths(string btnTxt, string resource_ButtonPath , string resource_PanelPath);
```

#### 9.3.3.1 PARAMETERS

|                     |        |   |
|---------------------|--------|---|
| btnTxt              | string | <i>Optional.</i> Sets the text for the button of the new tab. Default is “TabButton”. |
| resource_ButtonPath | string | <i>Optional.</i> Path of a button to use in a Resources folder.                       |
| resource_PanelPath  | string | <i>Optional.</i> Path of a panel to use in a Resources folder.                        |

This function uses the Insert Tab function to add a tab to the beginning of the tab list. It takes two optional ResourcePaths to GameObjects which is uses for the new tab’s button and panel.

---

### 9.3.4 ADDTABEND

```
public void AddTabEnd(string btnTxt);
```

#### 9.3.4.1 PARAMETERS

|        |        |   |
|--------|--------|---|
| btnTxt | string | <i>Optional.</i> Sets the text for the button of the new tab. Default is “TabButton”. |
|--------|--------|---|

This function uses the InsertTab function to add a tab to the end of the tab list.

---

### 9.3.5 ADDTABENDWITHOBJECTS

```
public void AddTabEndWithObjects(string btnTab, GameObject resource_ButtonPrefab ,
GameObject resource_PanelPrefab);
```

#### 9.3.5.1 PARAMETERS

|                       |            |   |
|-----------------------|------------|---|
| btnTxt                | string     | <i>Optional.</i> Sets the text for the button of the new tab. Default is “TabButton”. |
| resource_ButtonPrefab | GameObject | <i>Optional.</i> Uses this object for the new tab’s button.                           |
| resource_PanelPrefab  | GameObject | <i>Optional.</i> Uses this object for the new tab’s panel.                            |

This function uses the InsertTab function to add a tab to the end of the tab list. It takes two optional GameObjects which is uses for the new tab’s button and panel.

---

### 9.3.6 ADDTABEND

```
public void AddTabEndWithResourcePaths(string btnTxt, string resource_ButtonPath , string
resource_PanelPath);
```

#### 9.3.6.1 PARAMETERS

|                     |        |   |
|---------------------|--------|---|
| btnTxt              | string | <i>Optional.</i> Sets the text for the button of the new tab. Default is “TabButton”. |
| resource_ButtonPath | string | <i>Optional.</i> Path of a button to use in a Resources folder.                       |
| resource_PanelPath  | string | <i>Optional.</i> Path of a panel to use in a Resources folder.                        |

This function uses the InsertTab function to add a tab to the end of the tab list. It takes two optional ResourcePaths to GameObjects which is uses for the new tab’s button and panel.

---

### 9.3.7 COUNTACTIVEPANELS

```
public int CountActivePanels();
```

Returns the number of tab panels which are active. This is mainly used by TabInspector classes for error checking during editing. There should only ever be one active tab panel at a time.

---

### 9.3.8 INSERTTAB

```
public static GameObject InsertTab(int tabNum, string btnTxt, bool switchToNewTab);
```

#### 9.3.8.1 PARAMETERS

|                |        |  |
|----------------|--------|--|
| tabNum         | int    | The index of where to put the new tab.   |
| btnTxt         | string | <i>Optional.</i> Sets the text for the button on the new tab. Default is “TabButton”. If it is equal to “”, then no text element shall be added to the button. |
| switchToNewTab | bool   | <i>Optional.</i> Sets whether or not to switch to a the new tab. Default is the SWITCH_TO_NEW_TAB constant.  |

Creates a new tab at the specified index.

---

### 9.3.9 INSERTTABWITHOBJECTS

```
public static GameObject InsertTab(int tabNum, GameObject prefab_Button, GameObject
prefab_Panel string btnTxt, bool switchToNewTab);
```

---

#### 9.3.9.1 PARAMETERS

|                |            |  |
|----------------|------------|--|
| tabNum         | int        | The index of where to put the new tab.   |
| btnTxt         | string     | <i>Optional.</i> Sets the text for the button on the new tab. Default is "TabButton". If it is equal to "", then no text element shall be added to the button. |
| prefab_Button  | GameObject | <i>Optional.</i> Uses this object for the new tab's button.  |
| prefab_Panel   | GameObject | <i>Optional.</i> Uses this object for the new tab's panel.   |
| switchToNewTab | bool       | <i>Optional.</i> Sets whether or not to switch to a the new tab. Default is the SWITCH_TO_NEW_TAB constant.  |

Creates a new tab at the specified index. All functions which add a new tab use this function. It takes two optional GameObjects which is uses for the new tab's button and panel.

---

### 9.3.10 INSERTTABWITHRESOURCEPATHS

```
public static GameObject InsertTab(int tabNum, string prefabPath_Button = "", string
prefabPath_Panel = "", string btnTxt, bool switchToNewTab);
```

---

#### 9.3.10.1 PARAMETERS

|                   |        |  |
|-------------------|--------|--|
| tabNum            | int    | The index of where to put the new tab.   |
| btnTxt            | string | <i>Optional.</i> Sets the text for the button on the new tab. Default is "TabButton". If it is equal to "", then no text element shall be added to the button. |
| prefabPath_Button | string | <i>Optional.</i> Uses this string to find the resource for the new tab's button.   |
| prefabPath_Panel  | string | <i>Optional.</i> Uses this string to find the resource for the new tab's panel.  |
| switchToNewTab    | bool   | <i>Optional.</i> Sets whether or not to switch to a the new tab. Default is the SWITCH_TO_NEW_TAB constant.  |

Creates a new tab at the specified index. It takes two optional ResourcePaths to GameObjects which is uses for the new tab's button and panel.

---

### 9.3.11 MOVETABDOWN

```
public void MoveTabDown(int tabNum);
```

#### 9.3.11.1 PARAMETERS

|        |     |                               |
|--------|-----|-------------------------------|
| tabNum | int | The index of the tab to move. |
|--------|-----|-------------------------------|

This moves the tab down in the list by adding 1 to its SiblingIndex. This usually results in the tab moving beneath or to the right of other tabs. If it can't move the tab any further it will produce a debug log.

---

### 9.3.12 MOVETABUP

```
public void MoveTabUp(int tabNum);
```

#### 9.3.12.1 PARAMETERS

|        |     |                               |
|--------|-----|-------------------------------|
| tabNum | int | The index of the tab to move. |
|--------|-----|-------------------------------|

This moves the tab up in the list by subtracting 1 from its SiblingIndex. This usually results in the tab moving above or to the left of other tabs. If it can't move the tab any further it will produce a debug log.

---

### 9.3.13 REMOVEALLTABS

```
public void RemoveAllTabs();
```

Removes all the tabs in this TabContainer.

---

### 9.3.14 REMOVETAB

```
public void RemoveTab(int tabNum, bool forceRemoveLastTab);
```

#### 9.3.14.1 PARAMETERS

|                    |      |  |
|--------------------|------|--|
| tabNum             | int  | The index of the tab to remove.  |
| forceRemoveLastTab | bool | <i>Optional.</i> If true, it will remove the last tab if asked to. Default is false. |

Removes a tab at the specified index from the tab container. All functions which remove tabs use this function. By default, it does not delete the last tab in the container, as this could lead to problems. However, this can be overridden.

---

### 9.3.15 REMOVETABBEGIN

```
public void RemoveTabBegin();
```

Removed the tab at the beginning of the list via the RemoveTab function.

---

### 9.3.16 REMOVETABEND

```
public void RemoveTabEnd();
```

Removes the tab at the end of the list via the RemoveTab function.

---

### 9.3.17 RENUMBERTABS

```
public void ReNumberTabs();
```

Renumbers all the tabs based on their SiblingIndex. If the const FORCE\_RENAME is true, it also renames all the tab buttons and panels.

---

### 9.3.18 RESKIN [...] FUNCTIONS

```
public void ReskinAllButtons(string prefabPath_Button)
public void ReskinAllButtons(GameObject prefab_Button)
public void ReskinAllPanels(string prefabPath_Panel)
public void ReskinAllPanels(GameObject prefab_Panel)
public void ReskinAllTabs(string prefabPath_Button, string prefabPath_Panel)
public void ReskinAllTabs(GameObject prefab_Button, GameObject prefab_Panel)
public void ReskinButton(int tabNum, string prefabPath_Button)
public void ReskinButton(int tabNum, GameObject prefab_Button)
public void ReskinPanel(int tabNum, string prefabPath_Panel)
public void ReskinPanel(int tabNum, GameObject prefab_Panel)
public void ReskinTab(int tabNum, string prefabPath_Button, string prefabPath_Panel)
public void ReskinTab(int tabNum, GameObject prefab_Button, GameObject prefab_Panel)
```

All 12 reskin functions are very similar. They replace the text style and image of buttons and panels with those of a prefab or other GameObject. They accept 3 types of parameters; ints for specific buttons/panels/tabs, GameObjects which reference the object to base the new skin off of, and strings which point to a prefab in a Resources folder.

If you create your own button/panel, make sure the appropriate components are attached to it. You can use the default one in Plugins/Chimeforest/TabManager/Resources as a template.

---

### 9.3.19 SETLISTS

```
public void SetLists();
```

Populates the buttons and panels lists with the children of TabButtons and TabPanels GameObjects.

---

### 9.3.20 SETTABSINDEXVIAPANELS

```
public void SetTabsIndexViaPanels();
```

Sets the SiblingIndex to be equal to the Tab.tab of every tab button and panel. Kind of the opposite of ReNumberTabs. It is used by the TabInspector classes.

---

### 9.3.21 SWITCHTOTAB

```
public void SwitchToTab(int tabNum, bool recordUndo);
```

---

#### 9.3.21.1 PARAMETERS

|            |      |   |
|------------|------|---|
| tabNum     | int  | The index of the tab to switch to.  |
| recordUndo | bool | <i>Optional.</i> If true, while in the editor it will record an undo for switching tabs. Default is true. |

Sets a tab as the active tab.

---

### 9.3.22 TABPANELINDEXESMATCH

```
public bool TabPanelIndexesMatch();
```

Checks all the panel indexes and compares them to their Tab.tab. If one of them doesn't match, this returns a false, if all of them match, then it returns a true. Used by the TabInspector classes.

## 10 TABUTILMENU

Public class in Chimeforest.TabManager inherits from MonoBehavior

This class is mostly used as a hook for the TabUtilMenu, but it also stores a couple important variables.

### 10.1 VARIABLES

|            |        |            |   |
|------------|--------|------------|---|
| type       | public | string     | The type of object this script is attached to.            |
| TabManager | public | TabManager | A reference to the TabManager which controls this object. |