
WindowSystem Documentation

Release 1.02

kanemura

October 06, 2015

1 Feature	1
1.1 Features of window system	1
2 Tutorial:	2
2.1 Environment setting	2
2.2 This window system's samples	4
2.3 Tutorial	15
3 MulID,FiveCC:	24
3.1 Mul ID,FiveCC	24
4 Texture:	26
4.1 The texture	26
5 Sprite font	33
5.1 Font	33
6 Sound effect,BGM:	38
6.1 SoundEffect,BGM	38
7 String Resource,Multi language:	43
7.1 The multilingual	43
8 Window Script:	46
8.1 The window script(wra) syntax	46
9 Window:	49
9.1 An arrangement of the window	49
9.2 Window priority	49
9.3 WINDOW	51
10 Control:	62
10.1 An arrangement of the control	62
10.2 The control priority	64
10.3 The control size	66
10.4 Control common properties	66
10.5 TEXT	71
10.6 RICHTEXT	74
10.7 LOG	79
10.8 LOGTEXT	84
10.9 EDITBOX	87
10.10 TEXTBOX	91
10.11 BUTTON	96

10.12 RADIO	101
10.13 CHECKBOX	107
10.14 TEXTURE	112
10.15 LINE	116
10.16 RENDER	119
10.17 ICON	123
10.18 RECASTICON	127
10.19 RENDERICON	131
10.20 METER	135
10.21 SCROLLBAR	139
10.22 LISTBOX	144
10.23 LISTBOXEX	148
10.24 CONTAINER	153
10.25 FRAME	157
10.26 LABEL	160
10.27 BAR	164
10.28 CMainSystemBase	167
10.29 CAssetBundleMgr	169
10.30 About the AssetBundle version.	172
10.31 CTextureResourceMgr	173
10.32 CSoundEffectMgr	175
10.33 CSeResourceMgr	176
10.34 CSeResource	177
10.35 CBgmResourceMgr	177
10.36 CBgmResource	178
10.37 CMessageDataSheetMgr	178
10.38 CWindowMgr	180
10.39 CWindowBase	181
10.40 CWinCtrlBase	185
10.41 CWinContents	189
10.42 CWinCtrlText	190
10.43 CWinCtrlRichText	191
10.44 CWinCtrlLog	192
10.45 CWinCtrlLogText	194
10.46 CWinCtrlEditbox	195
10.47 CWinCtrlTextbox	195
10.48 CWinCtrlButton	196
10.49 CWinCtrlRadio	197
10.50 CWinCtrlCheckbox	198
10.51 CWinCtrlTexture	199
10.52 CWinCtrlLine	200
10.53 CWinCtrlRender	200
10.54 CWinCtrlIcon	202
10.55 CWinCtrlRecastIcon	202
10.56 CWinCtrlRenderIcon	204
10.57 CWinCtrlMeter	207
10.58 CWinCtrlScrollbar	208
10.59 CWinCtrlListbox	208
10.60 CWinCtrlListboxEx	210
10.61 CWinCtrlContainer	212
10.62 CWinCtrlFrame	212
10.63 CWinCtrlLabel	213
10.64 CWinCtrlBar	213
10.65 How to change the default value	214

Feature

1.1 Features of window system

The features of this window system is as follows.

Window-based system

By the window-based system, such as the screen transition is made easier. Since it can be stuck to the window together, it becomes easier to have return to the state before opening the window.

The system that does not collapse even if your application is developed by more than developers.

When your application is developed by more than developers, you use a version control software (git or svn) in many cases. In this case, many problems occur in the merge. The window system is able to solve these problems.

The separation of the window data and program logics

Powerful and simple window description language.

Script for describing the window definition can be easy and simple to write. It can be described radio buttons, tab controls, listbox, and etc. This script is able to write a window animation at open/close.

A flexible layout

Support a flexible layout not collapse even if the screen size is changed.

Development efficient texture atlas

Texture Atlas of the window system support is turned to what was one step evolution texture atlas with the Unity. You can specify a patch how individually when texture atlas. The patch is a scaling method of the texture. You can specify to extend only the middle and fix the both ends. In addition, when performing the Texture Atlas, it is possible to automatically color reduction to 16bit textures from 32bit. In this case, it is possible to set whether a simple color reduction or dithering color individually texture.

Optimized rendering

It has been to render together multiple controls in one of the mesh. Optimization of the draw call is automatically performed. Even a complex screen, it can provide a comfortable GUI.

Multilingual possibility for the window caption

The separation of the caption data and application. So the window system is able to make multilingual applications.

The window system is possible to separate the window editing environment and application building environment (It can be developed in a separate project). Thus, while modifying the window layout, it is possible to edit the application.

Tutorial:

2.1 Environment setting

2.1.1 For MS-Windows:Install the cygwin.

A window data need to compile the script. This compiler invokes a gcc's preprocessor.

Install the cygwin procedure in the MS-Windows environment as follows..

Download the cygwin

down load [setup-x86.exe](#) or [setup-x86_64.exe](#)

<http://www.cygwin.com/>

Installation by choosing the “Devel” package

In the “Select Packages”, the Devel **Default** → **Install** .

Install

Install as instructed.

Add the PATH of enviroment variable .

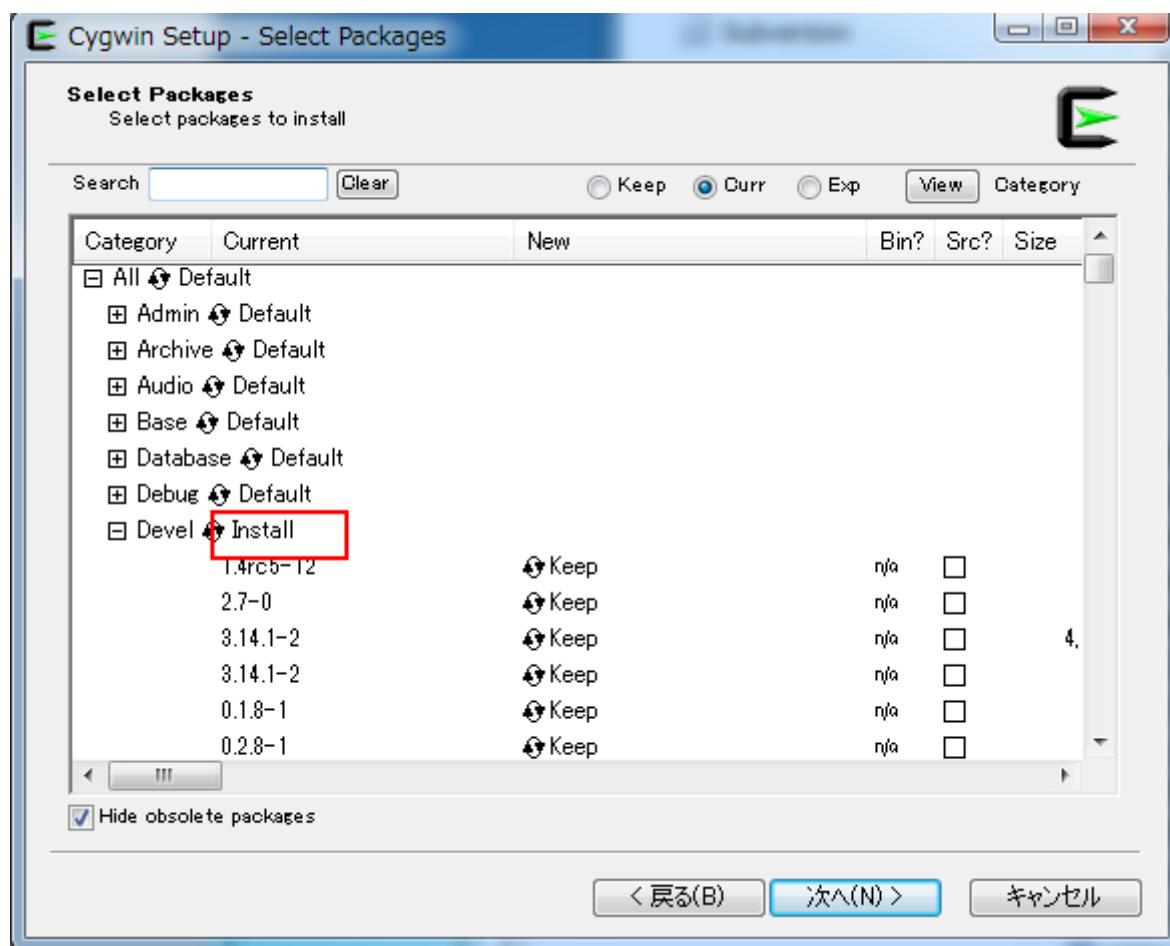
set the path to “bin” of the installed folder .

It has become in the following folder by default

- C:/cygwin/bin (if 64bit version, c:/cygwin64/bin)

The confirmation

Open a command prompt, try typing “gcc”. If you have been installed,it is displayed as follows.



```
C:\Windows\system32\cmd.exe
C:\Users>gcc
gcc: fatal error: no input files
compilation terminated.

C:\Users>
```

2.1.2 Install the Python 2.x

If you make a character resources for multi-lingual , use MS-Excel.This tool for data conversion from a Excel file is written in Python.

Library to read Excel file uses a **openpyxl**. Please install their own.

This environment has taken the operation check in Python 2.6 or later.

2.1.3 For OSX:Install the Command Line Tools in Xcode.

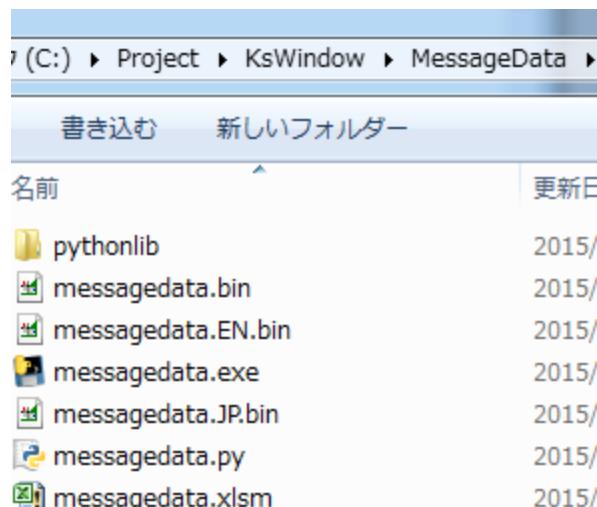
If you can not find the gcc on the terminal, please install the Command Line Tools in Xcode.

2.1.4 A preparation for the multi-lingual

You need to install the tools to convert to a binary file from Excel file for message data.

Please download it from here .

1. Expand the zip file to a top folder on your Unity Project.
2. The “pythonlib” folder is added as a library path of python (see python manual).



In this example,”Project/KsWindow” is the top folder of UnityProject

2.2 This window system's samples

2.2.1 A preparation for run the sample

Compile “wra” files

Please be sure you have done the *environment setting* .

If you have done,Highlight “KsSoft/WindowResource/” folder, please select the [Right-Click] → [Export].

All “wra” files are compiled.

If errors occur, please review the *environment setting* .

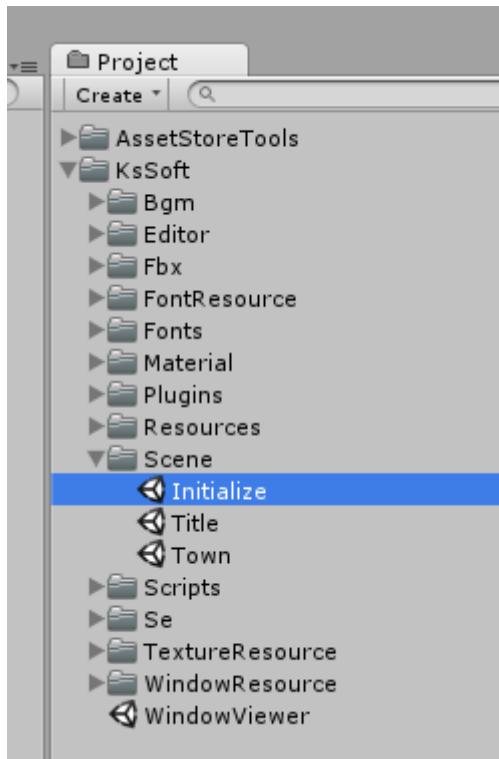
Output all of the asset bundle

From the [menu] [KsTools] → [Utility] → [Export All for Windows]

Note: If you try it in Mac environment, please confirm whether you install a command-line tool of the Xcode.

Run the sample scene

Run “KsSoft/Scene/Initialize.unity”.



2.2.2 The explanation of Sample Screens

The title

CWinTitle.cs

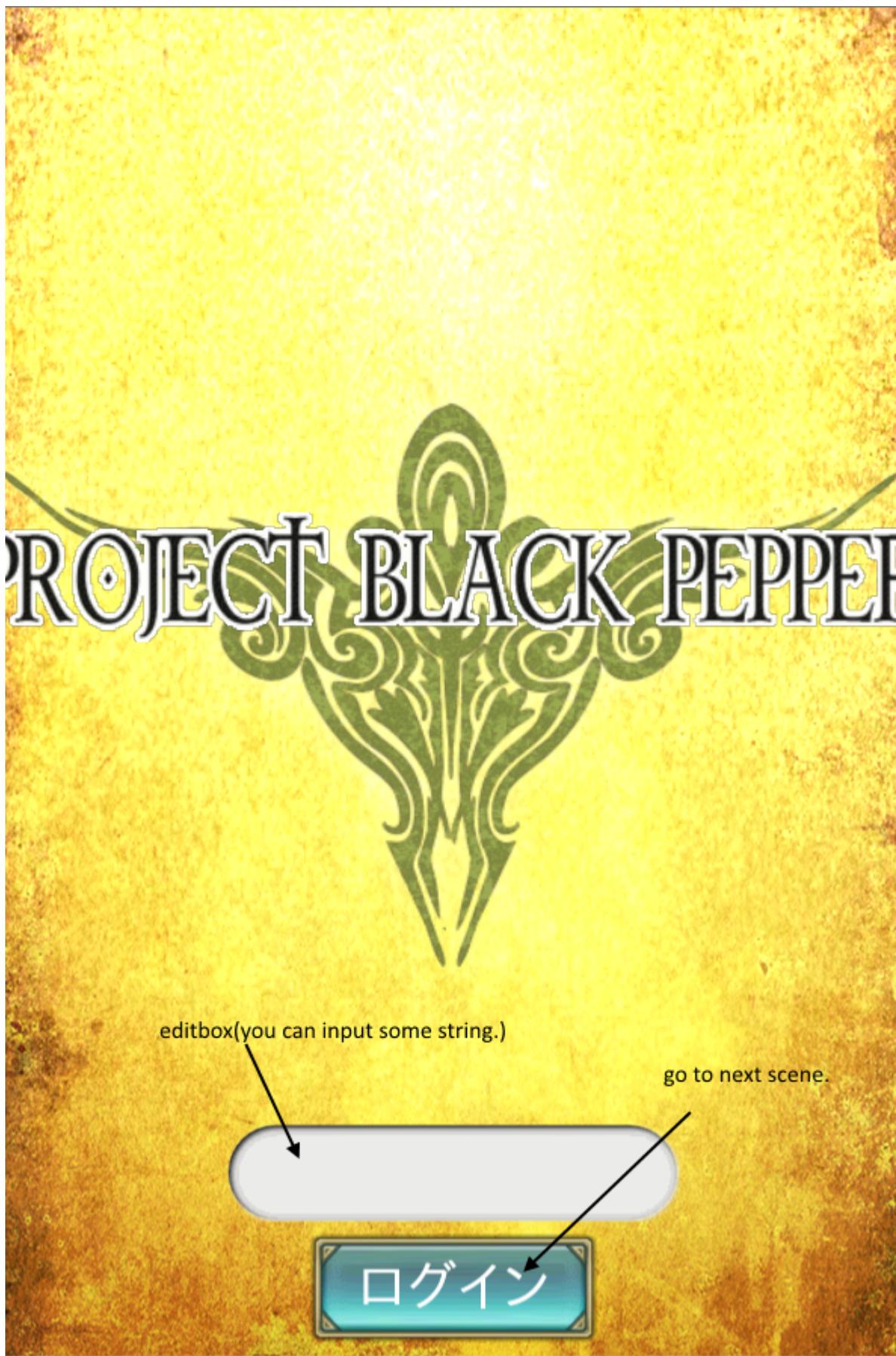
- KsSoft/Script/Scene/Title/CWinTitle.cs
- KsSoft/Script/Scene/Title/CWinTitleBase.cs
- KsSoft/WindowResource/CWinTitle.wra

The Home

CWinHome

- KsSoft/Script/Scene/Town/Home/CWinHome.cs
- KsSoft/Script/Scene/Town/Home/CWinHomeBase.cs
- KsSoft/WindowResource/CWinHome.wra

It is rendering the guild button, the friend button, the treasure button, and the center of the screen character. It is helpful on how to use the [RENDER](#).



CWinTopPart

- KsSoft/Script/Scene/Town/CWinTopPart.cs
- KsSoft/Script/Scene/Town/CWinTopPartBase.cs
- KsSoft/WindowResource/CWinTopPart.wra

It is responsible for the character status and render icon part of the top of the screen.

It is helpful on how to use the *RENDERICON*.

CWinBottomPart

- KsSoft/Script/Scene/Town/CWinBottomPart.cs
- KsSoft/Script/Scene/Town/CWinBottomPartBase.cs
- KsSoft/WindowResource/CWinBottomPart.wra

It is responsible for the chat button and the toggle part of the menu button at the bottom of the screen.

CWinTabbar

- KsSoft/Script/Scene/Town/CWinTabbar.cs
- KsSoft/Script/Scene/Town/CWinTabbarBase.cs
- KsSoft/WindowResource/CWinTabbar.wra

It is responsible for the menu icon part of the bottom of the screen.

The chat window

CWinChat

- KsSoft/Script/Chat/CWinChat.cs
- KsSoft/Script/Chat/CWinChatBase.cs
- KsSoft/WindowResource/CWinChat.wra

CWinChatInput

- KsSoft/Script/Chat/CWinChatInput.cs
- KsSoft/Script/Chat/CWinChatInputBase.cs
- KsSoft/WindowResource/CWinChatInput.wra

CWinChatPerson

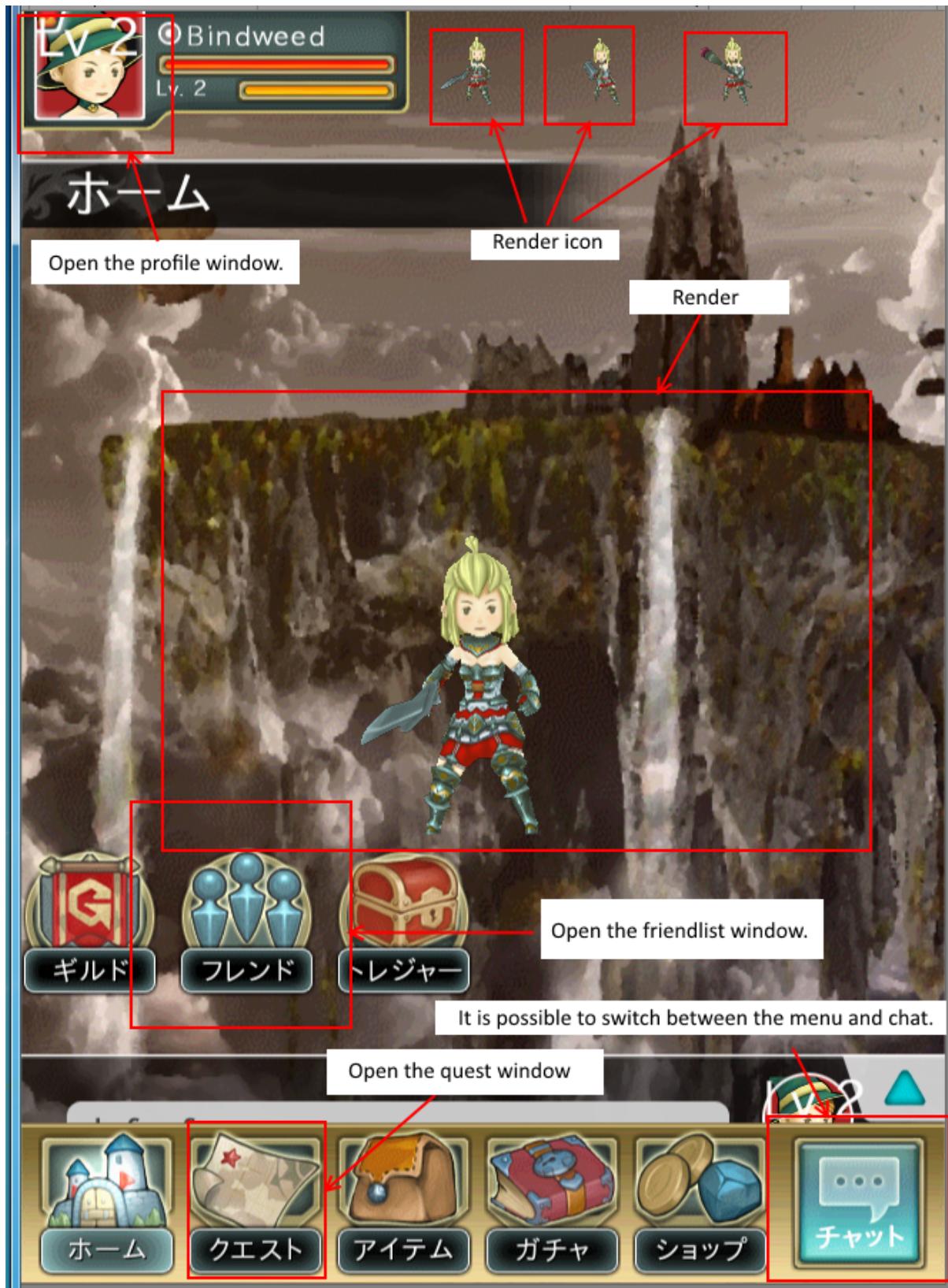
- KsSoft/Script/Chat/CWinChatPerson.cs
- KsSoft/Script/Chat/CWinChatPersonBase.cs
- KsSoft/WindowResource/CWinChatPerson.wra

This is an example of how to make the chat window. It is a sample of how to resizing the log window and displays the speaker icon.

The profile window

CWinProfile

- KsSoft/Script/Scene/Town/Profile/CWinProfile.cs
- KsSoft/Script/Scene/Town/Profile/CWinProfileBase.cs





- KsSoft/WindowResource/CWinProfile.wra

This is examples of horizontal list box. It is possible to page forward by swiping. In addition, it would be helpful on how to use the *RENDER*.



The friend list window

CWinFriend, CWinFriendList, CWinFriendApply, CWinFriendBalcklist

- KsSoft/Script/Scene/Town/Friend/CWinFriend.cs
- KsSoft/Script/Scene/Town/Friend/CWinFriendBase.cs
- KsSoft/WindowResource/CWinFriend.wra

- KsSoft/Script/Scene/Town/Friend/CWinFriendList.cs
- KsSoft/Script/Scene/Town/Friend/CWinFriendListBase.cs
- KsSoft/WindowResource/CWinFriendList.wra
- KsSoft/Script/Scene/Town/Friend/CWinFriendApply.cs
- KsSoft/Script/Scene/Town/Friend/CWinFriendApplyBase.cs
- KsSoft/WindowResource/CWinFriendApply.wra
- KsSoft/Script/Scene/Town/Friend/CWinFriendBalcklist.cs
- KsSoft/Script/Scene/Town/Friend/CWinFriendBalcklistBase.cs
- KsSoft/WindowResource/CWinFriendBalcklist.wra

This is an example of how to update the contents of the list box.

The friend list window

CWinQuest

- KsSoft/Script/Friend/CWinFriend.cs
- KsSoft/Script/Friend/CWinFriendBase.cs
- KsSoft/WindowResource/CWinFriend.wra
- KsSoft/Script/Friend/CWinFriendList.cs
- KsSoft/Script/Friend/CWinFriendListBase.cs
- KsSoft/WindowResource/CWinFriendList.wra
- KsSoft/Script/Friend/CWinFriendApply.cs
- KsSoft/Script/Friend/CWinFriendApplyBase.cs
- KsSoft/WindowResource/CWinFriendApply.wra
- KsSoft/Script/Friend/CWinFriendBalcklist.cs
- KsSoft/Script/Friend/CWinFriendBalcklistBase.cs
- KsSoft/WindowResource/CWinFriendBalcklist.wra

This is an example of how to update the contents of the list box.

The quest window

CWinQuest

- KsSoft/Script/Scene/Quest/CWinQuest.cs
- KsSoft/Script/Scene/Quest/CWinQuestBase.cs
- KsSoft/WindowResource/CWinQuest.wra

By giving content to the radio button, it is an example of how to behave it like a tab.







2.3 Tutorial

2.3.1 Scripting / Editing

1. Start your Unity Project.
2. You create a wra file under a “WindowResource” folder in the project(“Assets/KsSoft/WindowResource”).
3. Right-click and you select [Export] on the wra file
You write “wra” scripts with your favorite editor.
A file must be named as “window_name.wra”(ex: CMessageWindow.wra). The file encoding is UTF8.

The include path

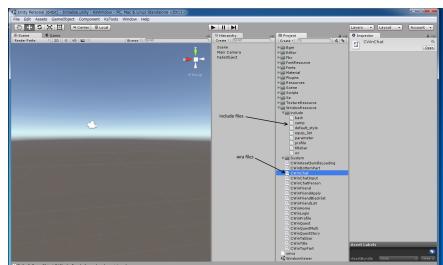
The include path is set to the following path.

Assets/KsSoft/WindowResource/include

The wra file that includ a “wr.h” automatically.

When you compile a “wra” file, “wr.h” is automatically included in your script. Common settings and constants are set to this file.

Assets/KsSoft/WindowResource/include/wr.h



The location of the compiled file.

*.wrb files are created under the folder “Unity Project/wrb”. These files are the result of the compiled your script files.

2.3.2 Window viewer

You can just preview the compiled windows in the viewer.

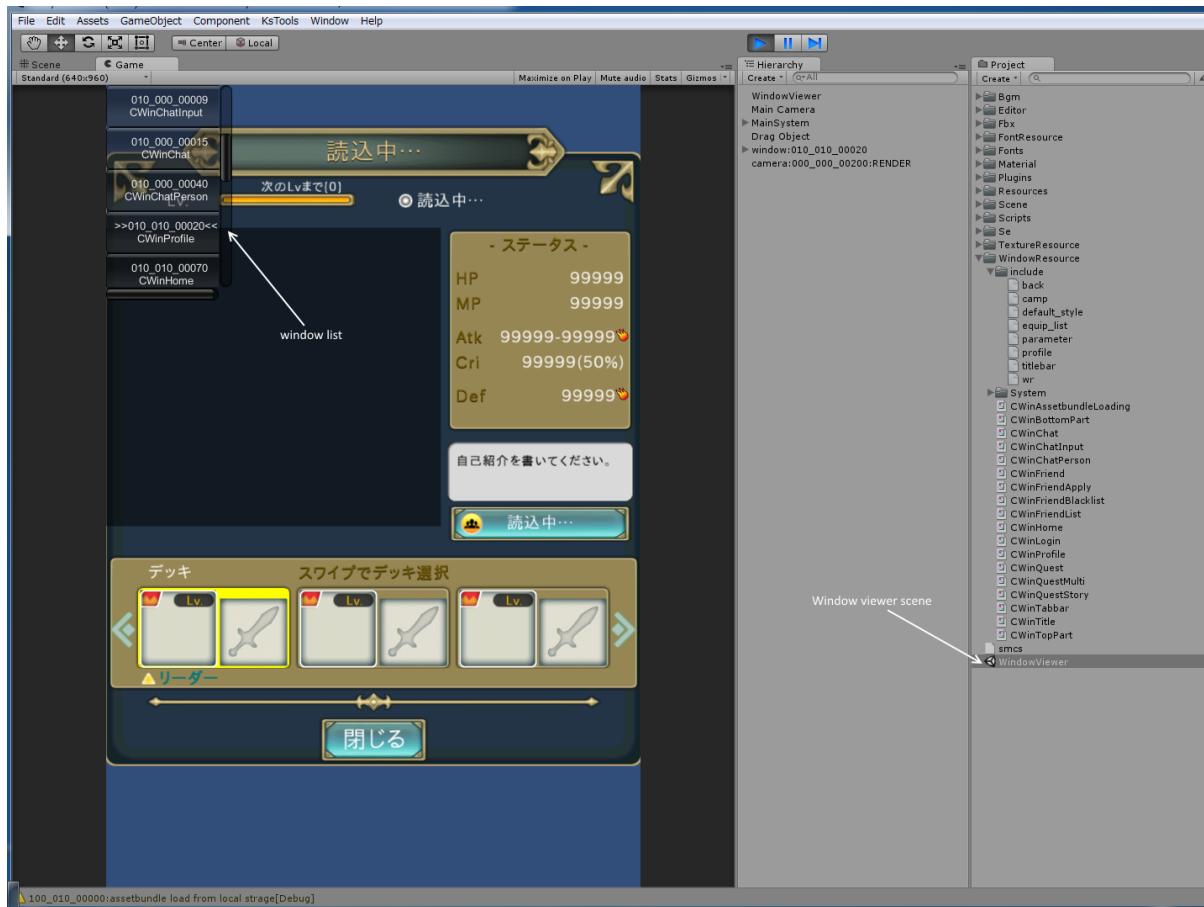
In addition, you can preview changes that the window was successful to compile once without stopping the viewer.

However, if you want to see the window that you have added a new, please restart once to end the window viewer.

1. Select the WWindowViewer scene, and run.

You can compile a your window script in the viewer scene running.

2. Since the list is displayed, choose a window that you want to preview.
3. The compilation window displays.



2.3.3 Assetbundles/Resources

It is possible to download the window data via HTTP.

Select the [KsTools]->[Export Window Resource] menu on the Unity. An assetfile is output to the “assetbundles” folder.

The assebundle file name is **000_010_00000.unity3d** as default.

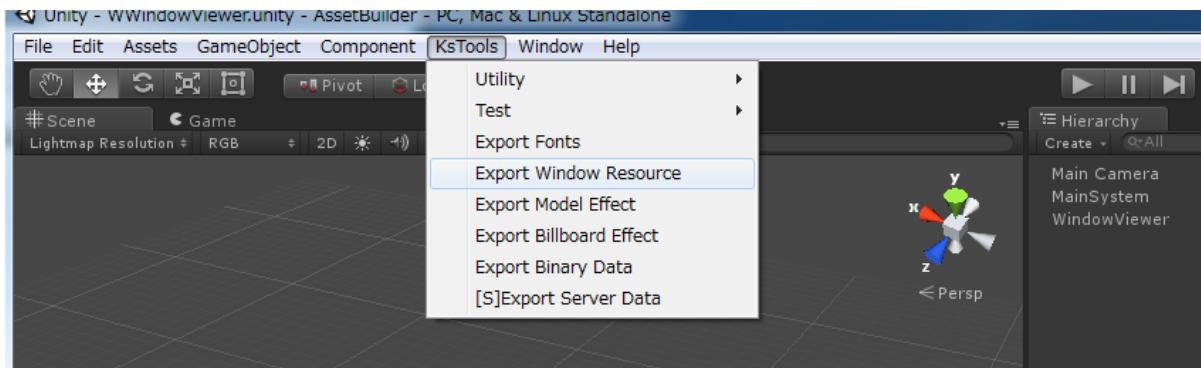
If you want to change the file to load, please refer to [here](#).

Using the **RESOURCE = MulID** of the WINDOW properties, it is possible to change the output file name.

Window data specifying the same **MulID** outputs by combining as a single asset. At the same time, it also updates ** version.unity3d ** storing information asset bundle.

CAAssetBundleMgr loads once ** version.unity3d ** at startup, asset bundle to check whether it has been updated.

It is also possible to output as an asset data using the WINDOW properties RESOURCE. The window data specifying the same path are output in a state of collectively as a single asset bundle.



2.3.4 How to handle the window in the app.

It can be handled in the same manner regardless to separate/not separate the window creation environment and application.

Create/Edit CMainSystem

```
public class CMainSystem : CMainSystemBase {
    //=====
    /*!Awake
     * @brief      Unity Callback
     */
    new void Awake() {
        Application.targetFrameRate = 60;
        base.Awake();

        if (m_instance != null) {
            Debug.LogError("already exist CMainSystem");
            return;
        }
        m_instance = this;

        // Add Component
        gameObject.AddComponent<CInput>();
        gameObject.AddComponent<CSpriteFontMgr>();
        gameObject.AddComponent<CTextureResourceMgr>();
        gameObject.AddComponent<CWindowMgr>();
    }
    //=====
    /*!初期化を行う。
     * @brief      initialize
     */
    override protected void initialize() {
        base.initialize();
    }
    //=====
    /*!Instance.
     * @brief      Instance.
     */
    static private CMainSystem m_instance = null;
    new public static CMainSystem Instance {
        get {

```

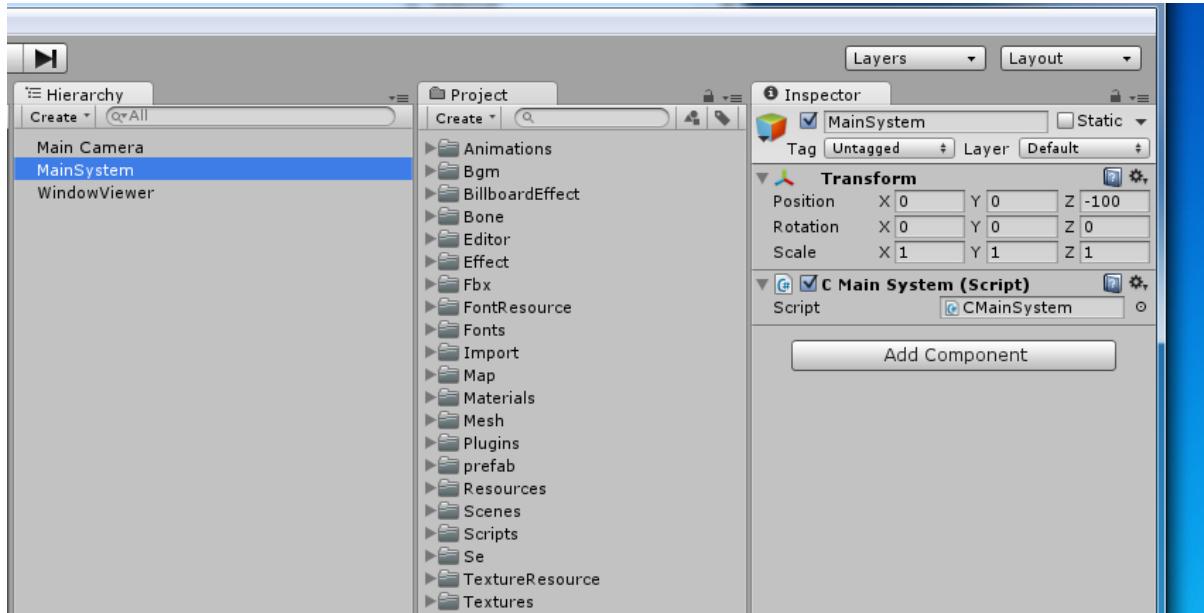
```

        return m_instance;
    }
}

```

Register the CMainSystem the game object.

As described below, to create an empty game object of that MainSystem, and add a CMainSystem as a script.



CMainSystem be created by inheriting the *CMainSystemBase*. Game objects add to CMainSystem is set so as not to be destroyed even if the transition to the scene (see DontDestroyOnLoad).

Scripts required to use the minimum window system is the four manager

For more details, refer to [here](#).

CInput

It is a script that monitors the input. This is necessary.

CSpriteFontMgr

It is managing the font data and font texture. This is necessary in order to render the character.

If you attach an CAssetBundleMgr the game object, it is possible to use to download the font data.

For more details, refer to [here](#).

CTextureResourceMgr

It manages the texture atlas resources. This is necessary.

If you attach an CAssetBundleMgr the game object, it is possible to use to download the texture data.

For more details, refer to [here](#).

CAssetBundleMgr

It manages the asset bundle.

It is used when loading an asset bundle data for sound effects, window layout data, texture resource data, font data, etc.

When not using the asset bundle, this is not necessary.

The manager is not concerned with the contents of the asset bundle.

For more details, refer to [here](#).

CWindowMgr

It manages the window. This is necessary.

For more details, refer to [here](#).

When you load from the resource

Load the assets were placed in the “Resources” folder.

Once the load, window data resides.

```
CWindowMgr cWindowMgr = CWindowMgr.Instance;
cWindowMgr.load("windows");
```

When you load from asset bundles

If you do [Export Window Resource], following two of the file is updated.(the case of the PC, Mac & Linux Standalone environment)

- assetbundles/Windows/000_000_00010.unity3d
- assetbundles/Windows/version.unity3d

Check whether [KsSoftConfig.httpserver](#) is pointing to the correct HTTP server folders.

Note: If you want to change the output file name ID, please refer to [here](#).

How to create a window

For example, it is assumed that defines the window in such a name

```
#include "default_style.h"

#define WIN_WIDTH 640
$y = 160;

WINDOW(001_000_00000) {
    RESOURCE = "Assets/KsSoft/Resources/windows";
    PATH = NETWORKPATH;
    TEX_ID = 100_000_00000;
    CAPTION = 000_000_0000;
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_NOFRAME|WINDOW_STYLE_ANCHOR_CENTER|WINDOW_STYLE_TOOLTIPS;
    SIZE = RELATIVE_SIZE(0),RELATIVE_SIZE(0);
    PRIORITY = PROGRESSBAR_PRIORITY;
};
```

```

METER(ProgressTotal) {
    ID = 000_001_00000;
    STYLE = ANCHOR_BOTTOM;
    POSITION = 0,$y;
    SIZE = RELATIVE_SIZE(-128);
    TEX_ID = 0,"MTRB";
    COLOR = COLOR32(255.0,255.0,255.0,255.0);
    SIZE1 = RELATIVE_SIZE(-128);
    TEX_ID1 = 0,"MTR";
    COLOR1 = COLOR32(255.0,255.0,255.0,255.0);
};

$y -=64;
METER(ProgressPart) {
    ID = 000_001_00010;
    STYLE = ANCHOR_BOTTOM;
    SIZE = RELATIVE_SIZE(-128),32;
    POSITION = 0,$y;
    TEX_ID = 0,"MTRB";
    COLOR = COLOR32(255.0,255.0,255.0,255.0);
    SIZE1 = RELATIVE_SIZE(-128),32;
    TEX_ID1 = 0,"MTR";
    COLOR1 = COLOR32(255.0,255.0,255.0,255.0);
};

TEXTURE(Wait) {
    ID = 000_002_00020;
    STYLE = ANCHOR_RIGHTBOTTOM;
    TEX_ID = 0,"LD00";
    POSITION = -48,32;
};

TEXT(Message) {
    ID = 000_002_00030;
    STYLE = TEXT_RIGHT|ANCHOR_RIGHTTOP;
    CAPTION = 000_000_00250;
    FONT_KIND = "cfn20";
    POSITION = -48,-32;
    COLOR = 1,1,1,1;
};

```

Then, CWinAssetbundleLoadingBase.cs(this is C# script) is generated.

You can create a window by calling the CWinAssetbundleLoadingBase.create () .

```
CWinAssetbundleLoading.create();
```

CWinAssetbundleLoadingBase that has been automatically generated have the following functions.

```

static public CWinAssetbundleLoading create(CWindowBase cParent = null) {
    return CWindowMgr.Instance.create<CWinAssetbundleLoading>(windowId,cParent);
}

```

How to get the windows that are create

The windows are create it can be obtained by using the A via window manager. You can get the window that is create by [find](#) .

```

CWinAssetbundleLoading cLoading = CWindowMgr.Instance.find<CWinAssetbundleLoading>(CWinAssetbundleLoading);
if (cLoading != null) {
    //クリエイト済のウィンドウが見つかった!
} else {
    //ウィンドウは、クリエイトしていない。
}

```

How to bring the window to the top.

you can bring the window to top by bringToTop.

```
CWindowMgr.Instance.bringToTop(cWindow);
```

About sound effects

Set the CWindowMgr.soundeffect To play the sound effect.

soundeffect is an object that has an interface that IWinSoundEffect.

```
public interface IWinSoundEffect {
    void play(uint mSE);
}
```

It is one of the examples to work with CSerResourceMgr.

You get the asset bundled been SE via *CSerResourceMgr* (It contains a multiple of SE to one of asset bundles).

CSerResource has a IWinSoundEffect interface.(To actually play the sound, it is necessary to add a *CSoundEffectMgr* to CMainSystem.)

In the example below, it assumed that you are pack the sound effect of the window system in 052_000_00000.

Note: It is also possible to associate the object with your own was IWinSoundEffect.

```
public class CMainSystem : CMainSystemBase {
    //=====
    /*!Awake
    * @brief Unity Callback
    */
    new void Awake() {
        base.Awake();

        if (m_instance != null) {
            Debug.LogError("already exist CMainSystem");
            return;
        }
        m_instance = this;

        // Add Component
        gameObject.AddComponent<CInput>();
        gameObject.AddComponent<CAsetBundleMgr>();
        gameObject.AddComponent<CSpriteFontMgr>();
        gameObject.AddComponent<CTextureResourceMgr>();
        gameObject.AddComponent<CWindowMgr>();
        gameObject.AddComponent<CBgmResourceMgr>();
        gameObject.AddComponent<CSerResourceMgr>();
        gameObject.AddComponent<CSoundEffectMgr>();
    }
    //=====
    /*!Initialize.
    * @brief initialize
    */
    override protected void initialize() {
        base.initialize();
        //-----
        // WindowMgr initilaize.
        //-----
    }
}
```

```

CWindowMgr      cWindowMgr = CWindowMgr.Instance;
// Assign standard SES.
cWindowMgr.soundeffect = CSesResourceMgr.Instance.reference(new MulId(52, 0, 0)); //2D系;
cWindowMgr.clickSE = new MulId(52, 0, 20);
cWindowMgr.scrollSE = new MulId(52, 0, 110);
}
=====
/*! Instance.
@brief     Instance.
*/
static private CMainSystem m_instance = null;
new public static CMainSystem Instance {
    get {
        return m_instance;
    }
}
}
}
}

```

How to character resources to asset bundle and multilingual.

When converting from the caption ID to a string, it is converted through CWindowMgr.captiondata of interface.

captiondata is IWinCaptionData interface.

Multilingual is made possible by changing the string returned in response to the current locale.

```

public interface IWinCaptionData {
    string find(uint mCaptionId);
}

```

It is one of the example of to work with *CMessageDataSheetMgr* .

Since *CMessageDataSheet* has a IWinCaptionData interface, you can get the caption data in the code, such as the following.

```

public class CMainSystem : CMainSystemBase {
    =====
    /*! Awake
    * @brief Unity Callback
    */
    new void Awake() {
        base.Awake();
        if (m_instance != null) {
            Debug.LogError("already exist CMainSystem");
            return;
        }
        m_instance = this;

        // Add Component
        gameObject.AddComponent<CIInput>();
        gameObject.AddComponent<CAAssetBundleMgr>();
        gameObject.AddComponent<CSpriteFontMgr>();
        gameObject.AddComponent<CTextureResourceMgr>();
        gameObject.AddComponent<CWindowMgr>();
        gameObject.AddComponent<CBgmResourceMgr>();
        gameObject.AddComponent<CSesResourceMgr>();
        gameObject.AddComponent<CSoundEffectMgr>();

        addManager(new CMessageDataSheetMgr(Utility.getSystemLocale()));
    }
    =====
}

```

```

/*!Initialize
* @brief    initialize
*/
override protected void initialize() {
    base.initialize();
    //-----
    // WindowMgr initialize.
    //-----
    CWindowMgr      cWindowMgr = CWindowMgr.Instance;
    cWindowMgr.captiondata = CMessageDataSheetMgr.Instance.find(new FiveCC("WNDW"));
    // Assign standard SEs
    cWindowMgr.soundeffect = CSesResourceMgr.Instance.reference(new MulId(52,0,0));
    cWindowMgr.clickSE = new MulId(52,0,20);
    cWindowMgr.scrollSE = new MulId(52,0,110);
}
//=====
/*!Instance.
@brief     Instance.
*/
static private CMainSystem m_instance = null;
new public static CMainSystem Instance {
    get {
        return m_instance;
    }
}
}
}

```

Note: It is also possible to associate the object with your own was IWinCaptionData.

MulID,FiveCC:

3.1 Mul ID,FiveCC

It describes the ID format that is used in the window script .

3.1.1 Mul ID

A Multi- ID is 32 bit integer divided to 8bit,8bit and 16bit.

Described in the following format.

8bit_8bit_16bit

You can set the value in the following range.

000_000_00000 ~ 255_255_65535

However , please “000_000_00000” do not use.

```
000_010_00000
123_123_12345
123_456_78910 //invalid(It is beyond the range of value)
```

Access method from C#

You can use the **MulId** class on C#.

```
//Constructor
MulId(string sId);
MulId(uint upper,uint middle,uint lower);
MulId(uint val);

// プロパティ
uint Upper;      //Get the upper 8bit
uint Middle;     //Get the middle 8bit
uint Lower;      //Get the lower 16bit

//cast to uint
static implicit operator uint (MulId mulId);
```

3.1.2 FiveCC

The ID consists of ASCII characters within five characters. You can use ASCII characters as follows.

- 0~9

- a~z
- A~Z
- ?,_
- Space

Please be careful ‘?’ and ‘_’. Because these are encoded in the same numerical value.

```
"BTN00"
"BTN0?" = "BTN_"
"Test"
```

Access method from C#

You can use the **FiveCC** class on C#.

```
//Constructor
FiveCC(string sId);
FiveCC(uint upper,uint middle,uint lower);
FiveCC(uint val);

// Get the character pointed to by the index
public char this[int index];

//cast to uint
static implicit operator uint (FiveCC FiveCC);
```

Texture:

4.1 The texture

Please refer [here](#) to access the texture resources on C#.

This window system uses what summarizes some of the texture(texture atlas). And when do the texture atlas, the summarized texture automatically reduces color to 16 bit by dithering.

If you want, when do the texture atlas, the summarized texture automatically reduced color to 16bit with dithering.

It can be chosen whether to perform dithering for each part.

■ Texture part

It can also be called a “part”. This is part of the texture atlas.

■ The texture

When referred to as a texture in this document ,it is the summarized texture(texture atlas).In addition, in order to clearly separate the normal texture , sometimes referred to as a texture resource.

4.1.1 How to make a texture resources

1. Start your Unity Project.
2. Create a folder with [*MulID*](#) under the “TextureResource” folder in your project (“Assets/KsSoft/TextureResource”).
3. Place textures in the folder .
4. Highlight a folder, please select the [Right-Click] → [Export] (multiple can be selected).

Thus, the asset bundle named 100_000_00000.unity3d is generated under the “assetbundles” folder.

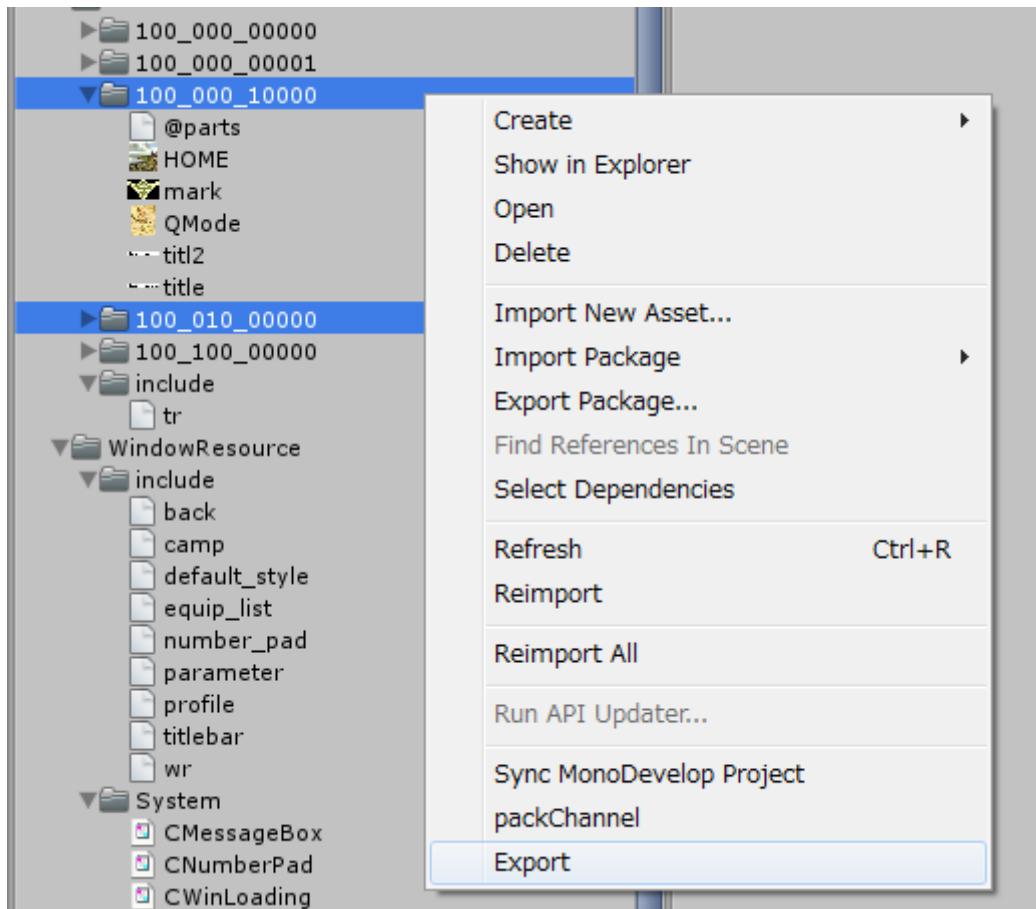
■ The texture name constraints

The name of the texture atlas must be 5 characters , except for the extension .

It is encoded as [*FiveCC*](#) .

Other, you can use the [*MulID*](#) file name.

BTN00.png
000_000_00010.png



■ The folder name constraints

It must be *MulID* .

Asset bundles because these are managed by *MulID* .

4.1.2 The definition file @parts.def

If a “@parts.def” placed in the texture folder, texture parts are converted along the definition file.

Because it is a simple text file , you can write it with your favorite editor.

The file encoding is UTF8.

Choose whether to convert to a resource.

```
RESOURCE = ON/OFF;
```

When you enable RESOURCE, It creates two files as follows under the “KsSoft/Resources/” folder.

- 001_000_00000.spr
- 001_000_00000.tex

By loading these two files , it is possible to load through the resources.

```
RESOURCE = ON;

PART("partA") {
    COLOR = 0.5,0.5,0,5,1;
};

PART("partB") {
```

```
COLOR = 0.5,0.5,0,5,1;
};
```

Texture Format

```
FORMAT = Texture Format;
```

The default texture format is **RGBA4444**.

Otherwise, it is possible to set a **PNG or JPG**.

Except when you want to specify a 16bit texture, please dithering to OFF.

```
DITHER = OFF;
FORMAT = RGBA32;
PART("partA") {
    COLOR = 0.5,0.5,0,5,1;
};
PART("partB") {
    DITHER = OFF;
    COLOR = 0.5,0.5,0,5,1;
};
```

How to texture atlas as PNG or JPG file.

It is possible to convert the texture atlas to a PNG or JPG files.

However, when it is loaded on the application, it is deployed as 32bit texture.

Please use it consider the advantages and disadvantages.

Format	Texture formats on application
FORMAT = PNG;	ARGB32
FORMAT = JPG;	RGB24

Advantage

Asset size decreases (= download size is smaller).

Disadvantages

Because it is deployed as a 32bit texture,it uses a lot of VRAM and main memory.

In addition, the rendering becomes heavier.

Note: On a narrow memory bandwidth smartphone, your application performance becomes degradation.

The default value for dithering

```
DITHER = ON/OFF;
```

Whether or not to the dithering can be set for each texture part.

When you do not specify, you can be selected either whether to default values.

In this example, partA is done the dithering, partB is not done.

```
DITHER = ON;
PART("partA") {
    COLOR = 0.5,0.5,0,5,1;
```

```
};

PART("partB") {
    DITHER = OFF;
    COLOR = 0.5,0.5,0,5,1;
};
```

Switching of the shader

```
SHADER = "SHADER PATH";
```

You can specify a shader for each texture resource.

You can not switch the shaders for each texture part.

But you can assign a shader to the texture resource.

```
SHADER = "Custom/Billboard";

PART("block") {
    COLOR = 1,1,1,1;
};
```

Texture part definitions and aliases

Texture part definitions can be defined in the form such as follows.

```
PART(Texture part file name) {
    Property 0;
    Property 1;
    :
    Property n;
};

PART(Alias name,Texture part file name) {
    Property 0;
    Property 1;
    :
    Property n;
};
```

You can assign a different name for one texture part. This makes it possible to act as parts with different properties in the same part.

For example, these are defined as follows. If you use “BTN00”, the window system renders it as it is. On the other hand, if you use “BTN01”, it is rendered dark.

```
PART("BTN00") {
    COLOR = 1,1,1,1;
};
PART("BTN01", "BTN00") {
    COLOR = 0.5,0.5,0.5,1;
};
```

The part definition properties

COLOR = R,G,B,A;

Change the color.

Elements of each color, specified in 0-1.

The default is 1, 1, 1, 1.

DITHER = ON/OFF;

It is possible to select whether or not dithering a part.

When the dithering is enabled, it is dithered with the “Jarvis, Judice, and Ninke” dithering.

The default is ON.

NODIVIDE;

NOPATCH;

A texture part is stretched equally.

Division of parts default has become NODIVIDE.

DIVIDE3H = Left fixed width,Right fixed width

PATCH3H = Left fixed width,Right fixed width

When a texture part is stretched, widths of left and right is fixed, and only the middle is stretched.

```
PART("BTN01") {
    DIVIDE3H = 24,24;
    DITHER = ON;
};
```

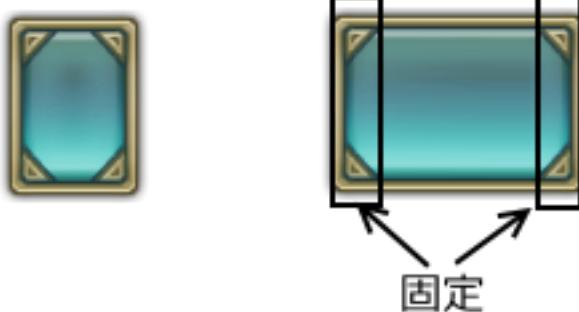


DIVIDE3V = Top fixed height,Bottom fixed height

PATCH3V = Top fixed height,Bottom fixed height

When a texture part is stretched, heights of top and bottom is fixed, and only the middle is stretched.

```
PART("BTN01") {
    DIVIDE3V = 24, 24;
    DITHER = ON;
};
```

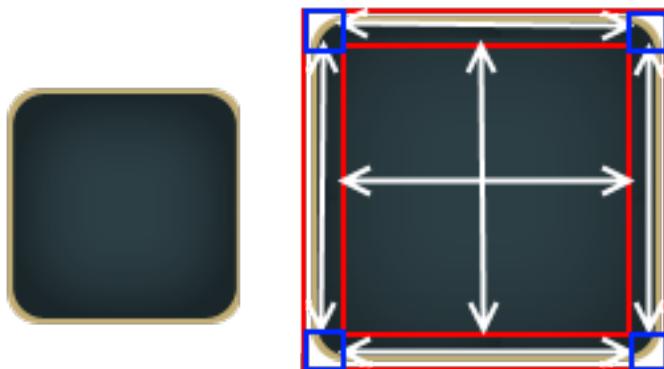


DIVIDE9 = Top left fixed width, Top left fixed height, Bottom right fixed width, Bottom right fixed height

PATCH9 = Top left fixed width, Top left fixed height, Bottom right fixed width, Bottom right fixed height

When a texture part is stretched, the four sides of the specified size is fixed and only the middle is stretched.

```
PART("FRAME") {
    DIVIDE9 = 40, 40, 40, 40;
    DITHER = OFF;
};
```



The include path

The include path is set to the following path.

```
Assets/KsSoft/TextureResource/include
```

@parts.def file that is automatically included

When you convert a texture resource, the “tr.h” is automatically included in “@part.def”.

```
Assets/KsSoft/TextureResource/include/tr.h
```

4.1.3 Preprocessor

You can use the same as C language preprocessor.

Preprocessor	Description
// Comment	Line comment
/* Comment */	Block comment
#include “file name”	Include the file
#define constant	Definition of a symbolic constant
#define Function macro	Function macro
#if defined(symbol definition) ~ #endif	Conditional compilation
#ifdef ~ #endif	Conditional compilation
#ifndef ~ #endif	Conditional compilation
#pragma once	Multiple include prevention

BuildTarget is defined as a macro.

```
#if defined(StandaloneWindows)
    MS-Windows
#else
    Other
#endif
```

Sprite font

5.1 Font

It can handle four bitmap fonts as one of the font texture.

When you convert these four bitmaps, it uses the α channel of the bitmap file. Four α channels are mapped to each channel(R,G,B,A) of a single texture.

If the number of bitmap exceeds 4, it can't be handled as a single font data.

5.1.1 Created as a resource data

Make a FontResource folder under the “Assets/KsSoft/” folder.

You put the text file and bitmap files that are named with *FiveCC* under the folder.

If you want to create a font named “cfn20”, you need to prepare the files such as follows. And you place these in this location.

- cfn20.txt
- cfn20_0.png
- cfn20_1.png
- cfn20_2.png
- cfn20_3.png

You highlight the “cfn20.txt”, and choose the [Right-click]→[Export]. Files such as the following are generated under the “Resources” folder.

- cfn20.font
- cfn20.tex

By these steps, your application can use the “cfn20” as a font data.

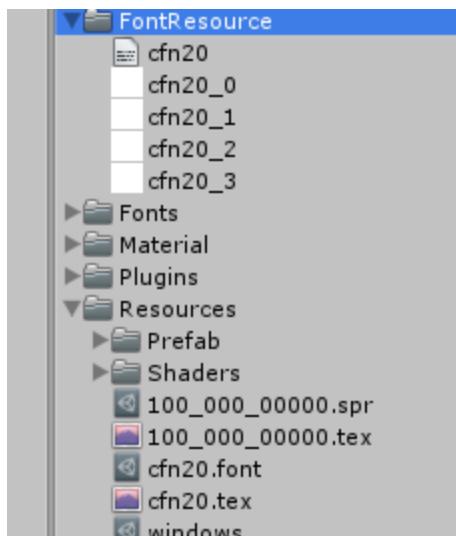
5.1.2 Created as an asset bundle data

Make a Fonts folder under the “Assets/KsSoft/” folder.

You put the text file and bitmap files that are named with *FiveCC* under the folder.

If you want to create a font named “fn16”, you need to prepare the files such as follows. And you place these in this location.

- fn16.txt
- fn16_0.png



- fn16_1.png
- fn16_2.png
- fn16_3.png

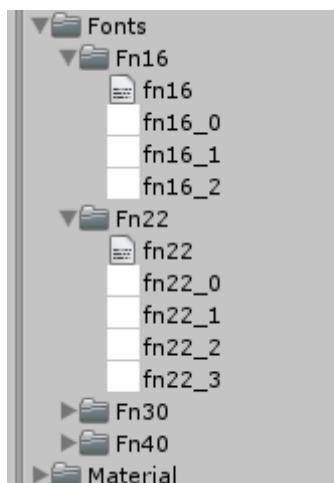
Also, when you want to use multiple fonts, put the number required by the same procedure.

Once you are ready, run the [KsTools]→[Export Fonts].

The following asset bundles are generated.

assetbundles/Windows/000_001_00000.unity3d

Within this asset bundles are stored all font data required.



Note: When you want to change the default value, please reference [here](#).

5.1.3 The font data generating method of BMFont

Introduce a method of making bitmap fonts by the BMFont(Bitmap Font Generator).

Creating a bitmap font

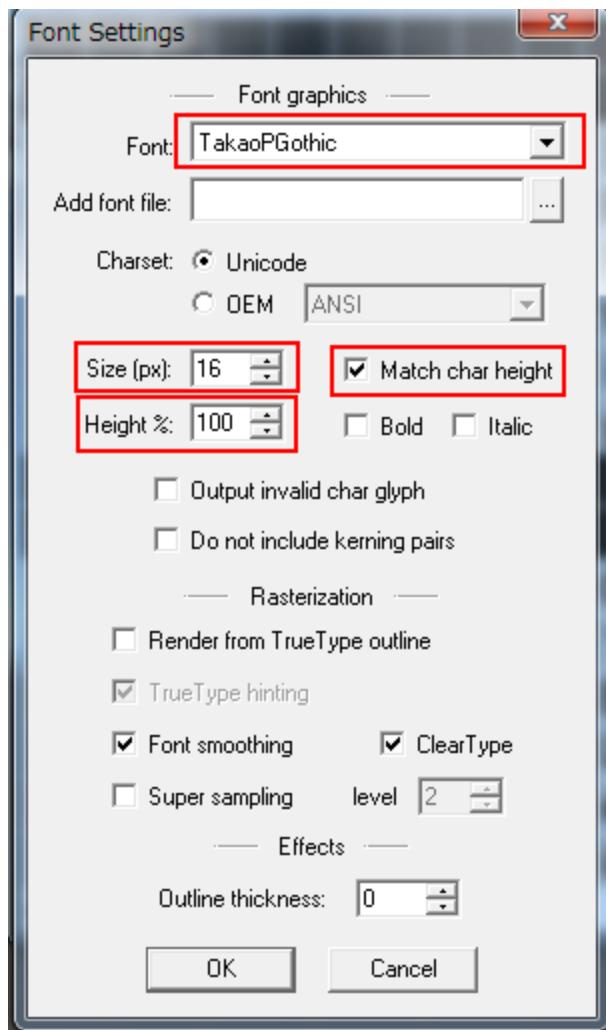
■ Font Settings

First, select the font you want to use. Select the [Options]→[Font Settings] from the menu.

Then select the font type and font size.

Put a check in the [Match char height].

Put 100 to [Hieght%].



■ Export Options

Next, to set the Export. Select the [Options] → [Export Options] from the menu.

You change the value of the part of the following red frame.

■ Select the character that you want to include

Then you select characters included in the bitmap font.

■ confirmation

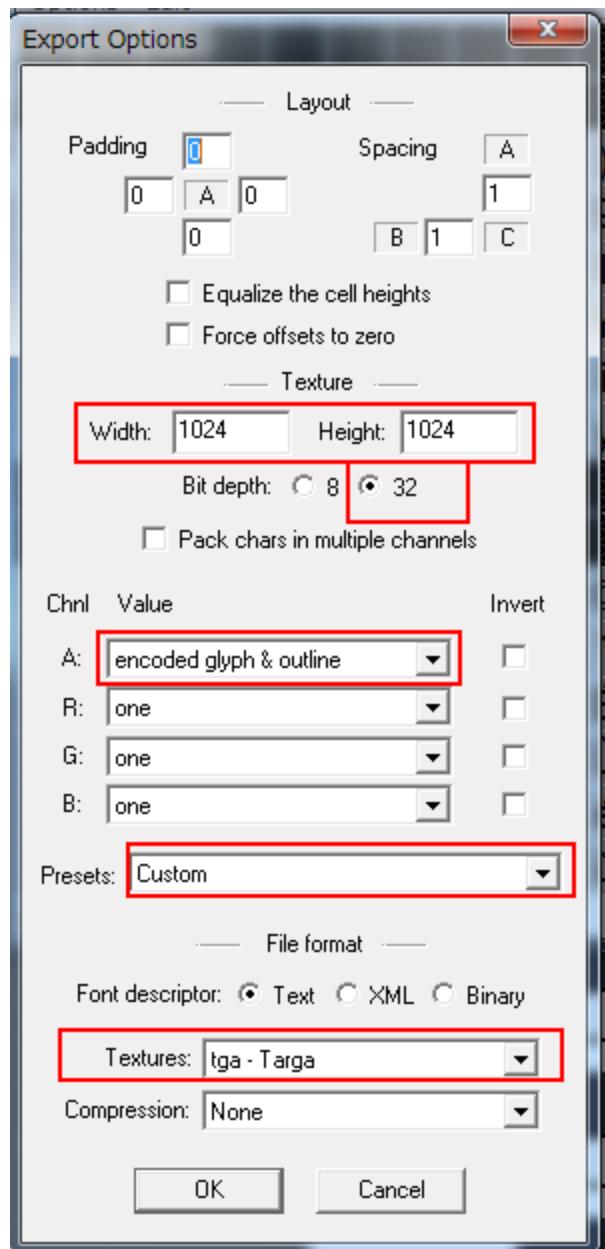
Ensure that you are correctly output along the items set. Select [Options] → [Visualize] from the menu, you can preview.

Make sure that bitmap number is within four.

If the bitmap number is one, almost always it may be possible to reduce the data size. You try to adjust the texture size that the bitmap number may be four.

■ Output

This configuration is complete.



Select the [Options] → [Save bitmap font as ...] from the menu, and then output bitmaps.

You need to save the file using a *FiveCC*.

Finally, you need to change the “fnt” extension to “fnt”.

Sound effect,BGM:

6.1 SoundEffect,BGM

Sound control managers are following .

- *CSoundEffectMgr*
- *CSeResourceMgr*
- *CBgmResourceMgr*

These manager control the management of asset bundles,SE grouping,priority and simultaneous sound limit. When playing the BGM,It is used in order to play the intro part + loop part .

Note: The SE grouping,priority, and simultaneous sound limit are supported by 3D sound only. The manager controls 2D SEs so as not to play the same sound at the same time.

6.1.1 How to create a Sound Effect asset bundle

To place an audio file to use for sound effects along to the next step.

1. Create a “Se” folder at the “Assets/KsSoft/” folder.
2. Under the “Se” folder you created, You create more folders in a *MulID* .
3. You place the audio files under the folder that you created in *MulID* name.
4. If necessary , also to create a file called @se.def.
5. Highlight a folder, please select the [Right-Click] → [Export] (multiple can be selected).

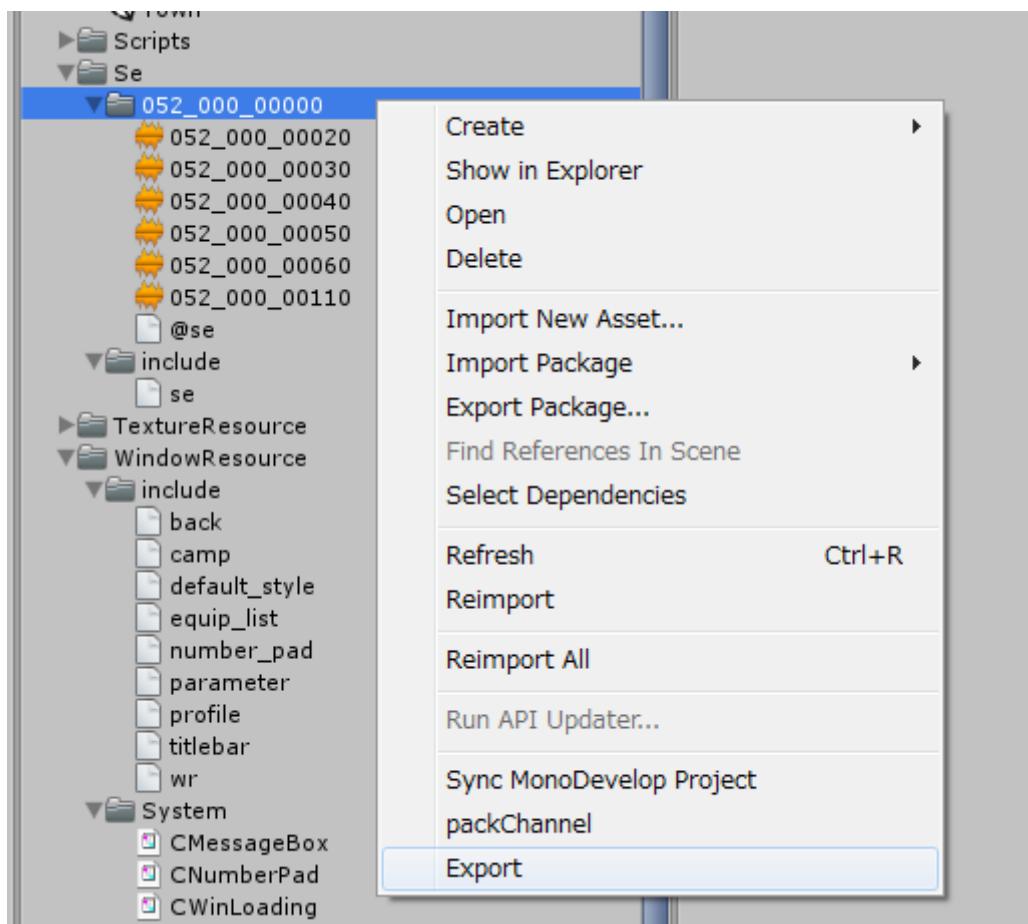
The folder name with the same name as the asset bundle is created.This file name is “assetbundles/XXXX/XXX_XXX_XXXXXX.unity3d”.

Placed sound files must be *MulID* or *FiveCC* .

6.1.2 @se.def syntax

When creating an asset bundle,you can embed various information to the SE.The following is this example. The file format is UTF-8(without BOM).

```
// click sound effect
SE(052_000_00020) {
    VOLUME = 1;
    PRIORITY = 1;
    GROUP = 200;
    POLYPHONY = 1;
```



```
};

// decide sound effect
SE(052_000_00030) {
    VOLUME = 50%;
    PRIORITY = 1;
    GROUP = 200;
    POLYPHONY = 1;
};
```

It is possible to provide an alias to the same audio and assign the different parameters.

```
SE(se file name) {
    Property 0;
    Property 1;
    :
    Property n;
};

SE(Alias name,SE file name) {
    Property 0;
    Property 1;
    :
    Property n;
};
```

Alias name / file name must be *MulID* or *FiveCC*.

It has the following syntax.

ID	File name
000_000_00010	000_000_00010.wav etc.
“click”	click.wav etc.

Note: When you use the *FiveCC* , please do not forget to enclose with the “~”

The include path

The include path is set to the following path.

```
Assets/KsSoft/Se/include
```

The @se.def that includ a “ se.h ” automaticaly.

When you compile a “@se.def” file,”tr.h” is automatically included in your script. Common settings and constants are set to this file.

```
Assets/KsSoft/Se/include/se.h
```

@se.def property

VOLUME = Number

Set the volume.

You can specified by a float of 0-1 , or you can use the 0-100% of the percentage.

- VOLUME = 1;

GROUP = Group number

It is effective only when a sound is a 3D.

Group number is specified by an integer of 0-255. It affects the simultaneous sound limit and priority.

PRIORITY = Priority number

It is effective only when a sound is a 3D.

The priority in the group set an integer of 0-255.

POLYPHONY = the simultaneous sound limit

It is effective only when a sound is a 3D.

Set between 1-255. It limits the number of simultaneous sound limit of the same group.If you exceed the number of simultaneous sound , this manager compare the sound priorities.If the priority which you want to play the sound is higher,it stop the sound that have lowest priority,and start to play new sound.

DISTANCE = minimum distance,maximum distance

It is effective only when a sound is a 3D.

When it plays at place nearer than the minimum distance , it plays at maximum volume.

In contrast , you can't hear the sound that the distance is more than the maximum.

Preprocessor

You can use the same as C language preprocessor.

Preprocessor	Description
// Comment	Line comment
/* Comment */	Block comment
#include "file name"	Include the file
#define constant	Definition of a symbolic constant
#define Function macro	Function macro
#if defined(symbol definition) ~ #endif	Conditional compilation
#ifdef ~ #endif	Conditional compilation
#ifndef ~ #endif	Conditional compilation
#pragma once	Multiple include prevention

BuildTarget is defined as a macro.

```
#if defined(StandaloneWindows)
    MS-Windows
#else
    Other
#endif
```

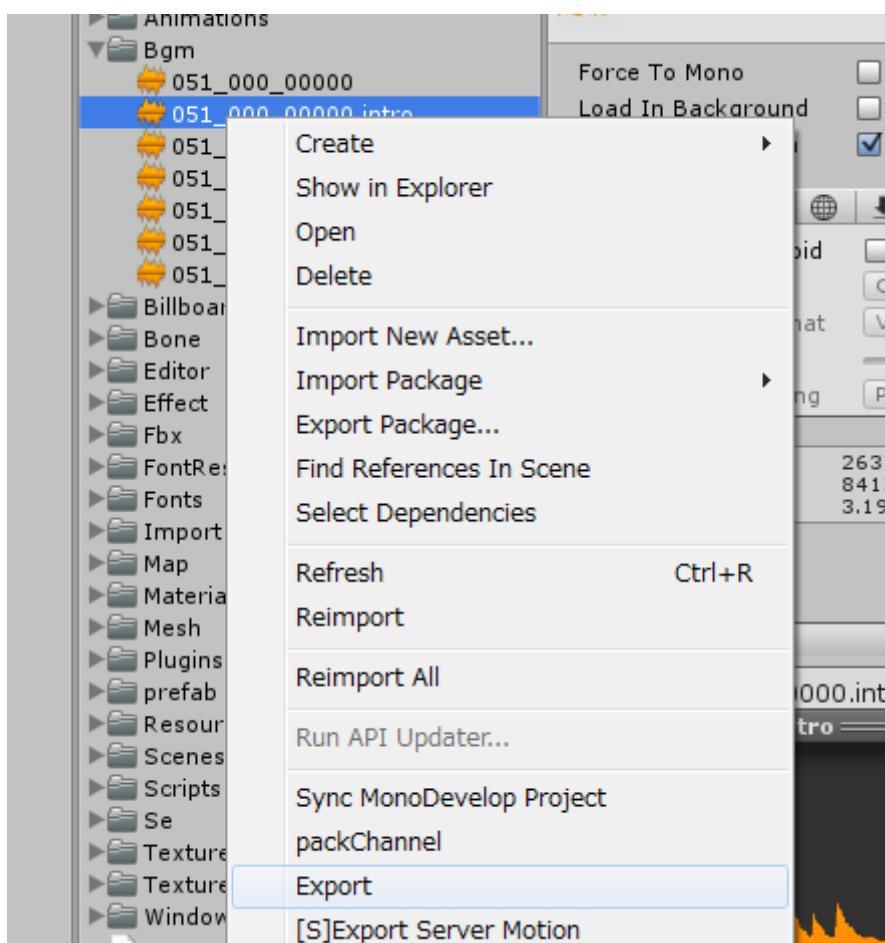
6.1.3 How to create an BGM's asset bundle

If necessary,BGM divide to the intro part and the loop part.You can create an asset bundle that were included them.The manager can be played automatically in order to read it.

It converted to asset bundle in the next procedure.

1. Create a “BGM” folder at the “Assets/KsSoft/” folder.
2. The loop part must be named “*MulID*.mp3”.
3. The intro part must be named “*MulID*.intro.mp3”.
4. Highlight the loop file or intro file that you want to convert into asset bundle.
5. [Right-click]→[Export]

The asset bundle named “*MulID*.unity3d” is generated.



String Resource,Multi language:

7.1 The multilingual

You can have the string resources as an asset data .

This system can change string resources depending on your application locale.

However , it is not essential on the window system .

When creating an asset data of the character data , using the MS-Excel.

We have prepared the tool for outputting binary data for the resource data from the Excel data.

C# Object for using the output data is are prepared as follows.

- *CMessagesheetMgr*
- *CMessagesheet*

7.1.1 The excel format of string resources

The excel file format is shown in the following figure .

[ORIGIN]

It parses the data on the basis of the column named [ORIGIN].

It is necessary to place it on the location where the data begins.

The same line as the [ORIGIN] will be commented. It does not affect the data output .

[D],[R]

[D] [R] of the B4 columns and C4 column select to whether or not to output data for what the version.
When individual column is blank , that line will suppress the data output.

- [D] is the debug version
- [R] is the release version

Character of D and R can be set freely. Whether or not to output any data by the argument can be selected. If you want to add new version, add the line.

	A	B	C	D	E
1	ウィンドウ用文字列データ			日本語データ	英語データ
2	[ORIGIN]	[D]	[R]	JP	EN
3		[D]	[R]	[D][R]	[D]
4					
5	● ● 000_000_00001	● ● 000_000_00002	● ● 000_000_00003	日本語データ	English
6	● ● 000_000_00004	● ● 000_000_00005	● ● 000_000_00006	名無しの権兵衛様	UnknownName
7	● ● 000_000_00007	● ● 000_000_00008	● ● 000_000_00009	閉じる	Close
8	● ● 000_000_00010	● ● 000_000_00011	● ● 000_000_00012	保存	Save
9	● ● 000_000_00013	● ● 000_000_00014	● ● 000_000_00015	戻る	Back
10	● ● 000_000_00016	● ● 000_000_00017	● ● 000_000_00018	進む	Continue
11				編集	Edit
12				新規	New
13				Lv.	Lv.
14				読み込中…	Loading…
15				名前	Name
16				レベル	Level
17				お気に入り	Favorite
18				日付	Date
19				パーティ	Party
20				HP	HP
21				MP	MP
22				攻撃力	Attack
	WNDW	QUES	FLOR	ITEM	MGEF
	YSYM	SKIL	TUTL		

ID

ID of D3 column is specify the string resource ID by the *MulID.[D][R]* of D4 represents that it is necessary for both the debug version and release version.

Since access to the character resources using the ID, It is required for all versions.

JP,EN

E3, F3 column of JP, EN is the locale name of string resources. Data is output as follows .

- messagedata.JP.bin
- messagedata.EN.bin

You can add any columns.

When your application locale is JP,[D][R] of E4 represents to outputting E column's data in both the release and debug.

On the other hand , [D] of F4 represents to outputting in the debug version only.

Sheet name [WNDW]

Sheet names are encoded in *FiveCC*.

There is a need to put the sheet name of up to 5 alphanumeric characters .

You can access the string resource by the argument of the sheet name and ID.

7.1.2 How to convert from Excel data to string resource data

You can use messagedata.exe or messagedata.py.

Installation please refer to [here](#).

This converter is written in python (2.6 or later).

It has imported the openpyxl to read Excel data.

If you want to convert to the exe file from this python script, You are possible to use the pyinstaller.

[How to Use] messagedata.exe or messagedata.py

messagedata [-v version] Excel file name [output file name]

- -v version

It outputs the data of the string set in version. By default, it has become D.

- Excel file name

Specify the Excel file.

- output file name

When it is omitted, it generates from the Excel file name.

```
messagedata -v R messagedata.xlsx caption

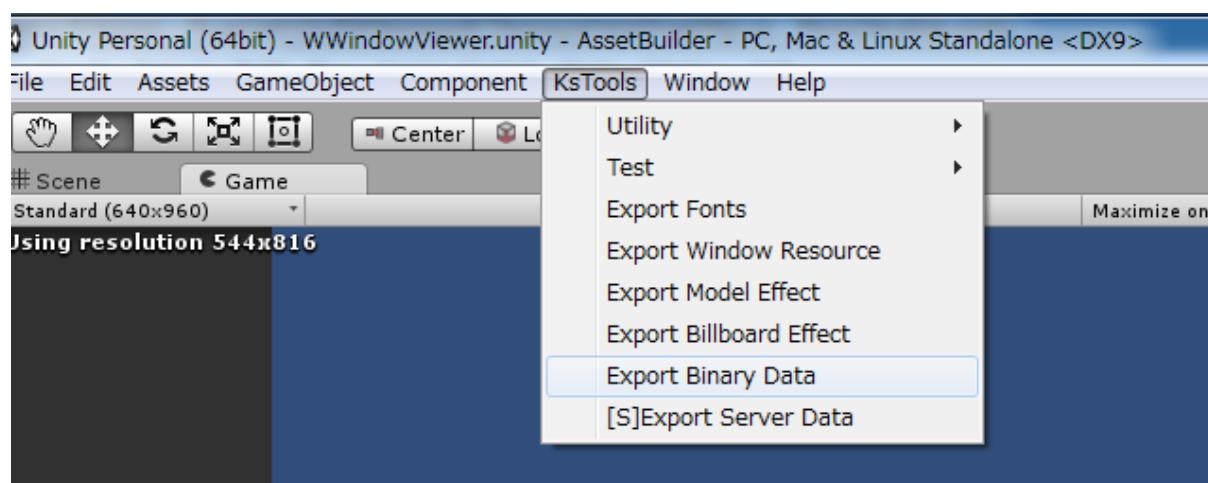
output file names:
messagedata.JP.bin
messagedata.EN.bin
etc
```

Asset bundling

This is the procedure of asset bundling that is provided as standard.

There is also a method which load a data as binary.

Select the [KsTools]→[Export Binary Data] from the menu.



If you want to add the locale data, add in the [KsSoftConfig](#).

Window Script:

8.1 The window script(wra) syntax

8.1.1 The wra syntax

a wra is composed of two main blocks.

- The window definition block
- Control definition blocks

Always , window definition block comes to the top. Following that, there is a control definition block.

The window-ID is required to be *MulID*.

```
// window define block
WINDOW(Window ID) {
    Property 0;
    Property 1;
    :
    Property n;
};

// control define blocks
ControlKind(name 0) {
    Property 0;
    Property 1;
    :
    Property n;
};
ControlKind(name 1) {
    Property 0;
    Property 1;
    :
    Property n;
};
:
ControlKind(name n) {
    Property 0;
    Property 1;
    :
    Property n;
};
```

```
#include "wr.h"
$y = 160;
#pragma RESOURCE ON

WINDOW(250_000_00010) {
```

```

RESOURCE;
PATH = NETWORKPATH;
ID = 001_000_00000;
TEX_ID = 100_000_00000;
CAPTION = 000_000_00000;
STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_NOFRAME|WINDOW_STYLE_ANCHOR_CENTER|WINDOW_STYLE_ANCHOR_BOTTOM;
SIZE = RELATIVE_SIZE(0),RELATIVE_SIZE(0);
PRIORITY = PROGRESSBAR_PRIORITY;
};

METER(ProgressTotal) {
    ID = 000_001_00000;
    STYLE = ANCHOR_BOTTOM;
    SIZE = RELATIVE_SIZE(-128);
    POSITION = 0,$y;
    TEX_ID = 0,"MTRB";
    COLOR = COLOR32(255.0,255.0,255.0,255.0);
    TEX_ID1 = 0,"MTR";
    COLOR1 = COLOR32(255.0,255.0,255.0,255.0);
};

$y -=64;
METER(ProgressPart) {
    ID = 000_001_00010;
    STYLE = ANCHOR_BOTTOM;
    SIZE = RELATIVE_SIZE(-128),32;
    POSITION = 0,$y;
    TEX_ID = 0,"MTRB";
    COLOR = COLOR32(255.0,255.0,255.0,255.0);
    TEX_ID1 = 0,"MTR";
    COLOR1 = COLOR32(255.0,255.0,255.0,255.0);
};

```

8.1.2 Variables and expressions

You can use the variables and the expression on the wra file. Those that begin from \$ is a variable.

The type of the variable is real. It is possible to write an expression in the property.

```
ID = 000_000_00010 + (3 * $val);
```

The declaration of variables

When you use a variable , it is necessary to always have an initial value. Uninitialized variables are errors.

```
$i = 3; //integer(Decimal number)
$h = 0xffff123; //integer(Hexadecimal number)
$f = 3.14; //real number
$m = 001_002_00003; //Multi (8bit,8bit,16bit)
```

Operators that can be used in the expression.

Operators	Description
+,-,*,/	Four arithmetic operations.
%	Surplus
,&,^,~	Bit operations(or, and, xor, not)
<<,>>	Shift operations
**	Exponentiation

The constant representation of the expression

Constants Examples	Description
14	Decimal integer
0x12fffff	Hexadecimal integer
000_001_00010	マルチ ID(8bit,8bit,16bit 形式)
0.345	Floating point

Calculation of string

a variable can't store a string. But, the operation for connecting the strings can be used.

- string + string
- string . string

```
PATH = "test/" + "folder/";
```

8.1.3 Preprocessor

You can use the same as C language preprocessor.

Preprocessor	Description
// Comment	Line comment
/* Comment */	Block comment
#include "file name"	Include the file
#define constant	Definition of a symbolic constant
#define Function macro	Function macro
#if defined(symbol definition) ~ #endif	Conditional compilation
#ifdef ~ #endif	Conditional compilation
#ifndef ~ #endif	Conditional compilation
#pragma once	Multiple include prevention

Next, list the preprocessor definitions of wra.

Preprocessor	Description
#pragma RESOURCE <i>MulID</i>	output as an asset bundles Example #pragma RESOURCE 000_014_000000
#pragma RESOURCE path	Output it to the specified path as resource data. Example #pragma RESOURCE = Resources/windows.asset
#pragma PATH "export path"	You want to change the output path. By default , the following folder. #pragma PATH ./wrc/base
#pragma BASECLASS Base class name	You want to change the base class By default , the following class #pragma BASECALSS CWindowBase

Window:

9.1 An arrangement of the window

9.1.1 The screen:

It is possible to define a virtual screen in the screen by setting the *SCREEN*.

Your window can adjust the center position and base position by *SCREEN* property.

A window position and RELATIVE_SIZE are calculated on the basis of the screen.

If you do not explicitly set it, the *SCREEN* property size is the same as your device screen size.

```
SCREEN = 0,0,RELATIVE_SIZE(0),RELATIVE_SIZE(0); //this mean is full screen.
```

By using the RELATIVE_SIZE, it can be set in a relative from the screen size.

```
STYLE = WINDOW_STYLE_ANCHOR_CENTER;
SCREEN = 0,-100,RELATIVE_SIZE(0),RELATIVE_SIZE(-150);
SIZE = 400,400;
```

In this case, it places the window as shown in the figure below.

9.1.2 Anchor:

When placing the window, you can set individually the origin position on the screen.

You can set the relative position of the window position from each origin position.

Even if a screen size is changed, the window layout is less likely to collapse.

We call it the anchor in this window system.

You can select nine kind anchor as shown in the figure below.

9.2 Window priority

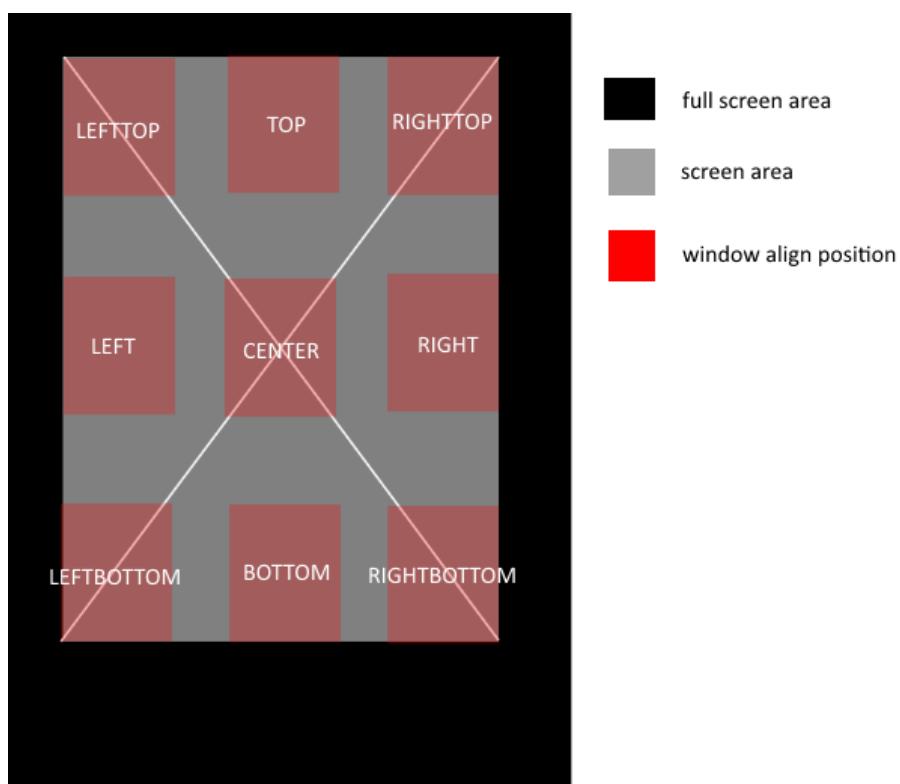
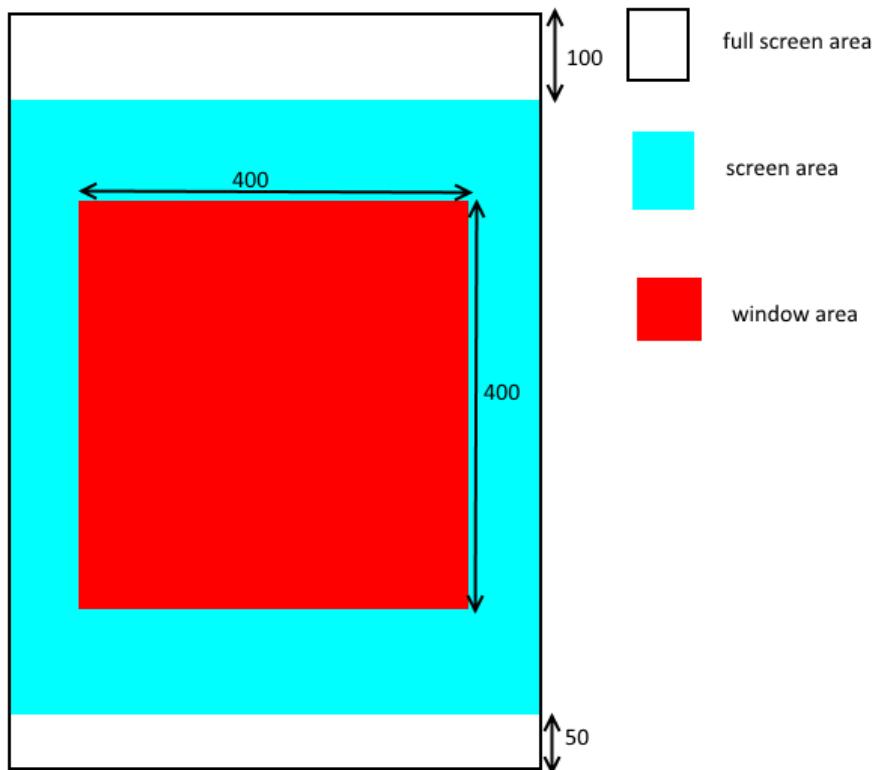
Generally, the newly created window is displayed above others.

If you want to control an explicit priority, it is necessary to set the value to *PRIORITY*.

The larger the priority, the window is displayed in front.

However, if windows are specified by the same style, you can control them using *PRIORITY*.

- TOP
- POPUP



- TOPMOST

However, if a window that specify the same style, you can control the priority using PRIORITY.

For example, if two windows are create by specifying the TOP, the priority of the window is determined by *PRIORITY*.

In addition, priority between TOP,POPUP and TOPMOST is as follows.

TOP < POPUP < TOPMOST

Window display priority flag	Description
WINDOW_STYLE_TOP	Set the display priority of the window to maximum. This is a lower priority than POPUP and TOPMOST. If the same priority flag has been specified,display priority is determined by the PRIORITY property.
WINDOW_STYLE_POPUP	Set the display priority of the window to maximum. It is priority than TOP, and a lower priority than TOPMOST. If the same priority flag has been specified,display priority is determined by the PRIORITY property. if you touch the screen other than this window, this window is closed.
WINDOW_STYLE_TOPMOST	Set the display priority of the window to maximum. This is a higher priority than TOP and POPUP. If the same priority flag has been specified,display priority is determined by the PRIORITY property.
WINDOW_STYLE_NOECLIPSE	Even if this window is opened, you specify it when you do not want to darken the other window.(valid only when you have specified the TOP / POPUP)

9.3 WINDOW

```
WINDOW(Window ID) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

Window ID is *MulID*.

If a wra file name is **TestWindow.wra**, the wra compiler automatically generates TestWindowBase.cs.

TestWindowBase.cs

Create a TestWindow.cs, and inherit the TestWindowBase.

The path for outputting the TestWindowBase.cs can be specified using the *PATH* on the script.

```
using UnityEngine;
using System;

public class TestWindow : TestWindowBase {
    :
    :
}
```

9.3.1 Example

Minimum window script

```
WINDOW(250_000_00000) {
    STYLE = WINDOW_STYLE_ANCHOR_CENTER;
    SIZE = 400, 400;
};
```

The following window is displayed in this script.

The origin position of the window is the upper left corner of the frame.(It is not the upper left corner of the title bar.)

Title bar will not be visible when you described as follows.

```
WINDOW(250_000_00000) {
    STYLE = WINDOW_STYLE_ANCHOR_TOP;
    SIZE = 400, 400;
};
```

Title bar will appear when you described as follows.

```
WINDOW(250_000_00000) {
    STYLE = WINDOW_STYLE_ANCHOR_TOP;
    POSITION = 0, -64;
    SIZE = 400, 400;
};
```

Window script without the closing button

You set a style as follows to delete the closing button.

```
WINDOW(250_000_00000) {
    STYLE = WINDOW_STYLE_NOCLOSE|WINDOW_STYLE_ANCHOR_CENTER;
    SIZE = 400, 400;
};
```

Window script without the title bar

Window script without the title bar

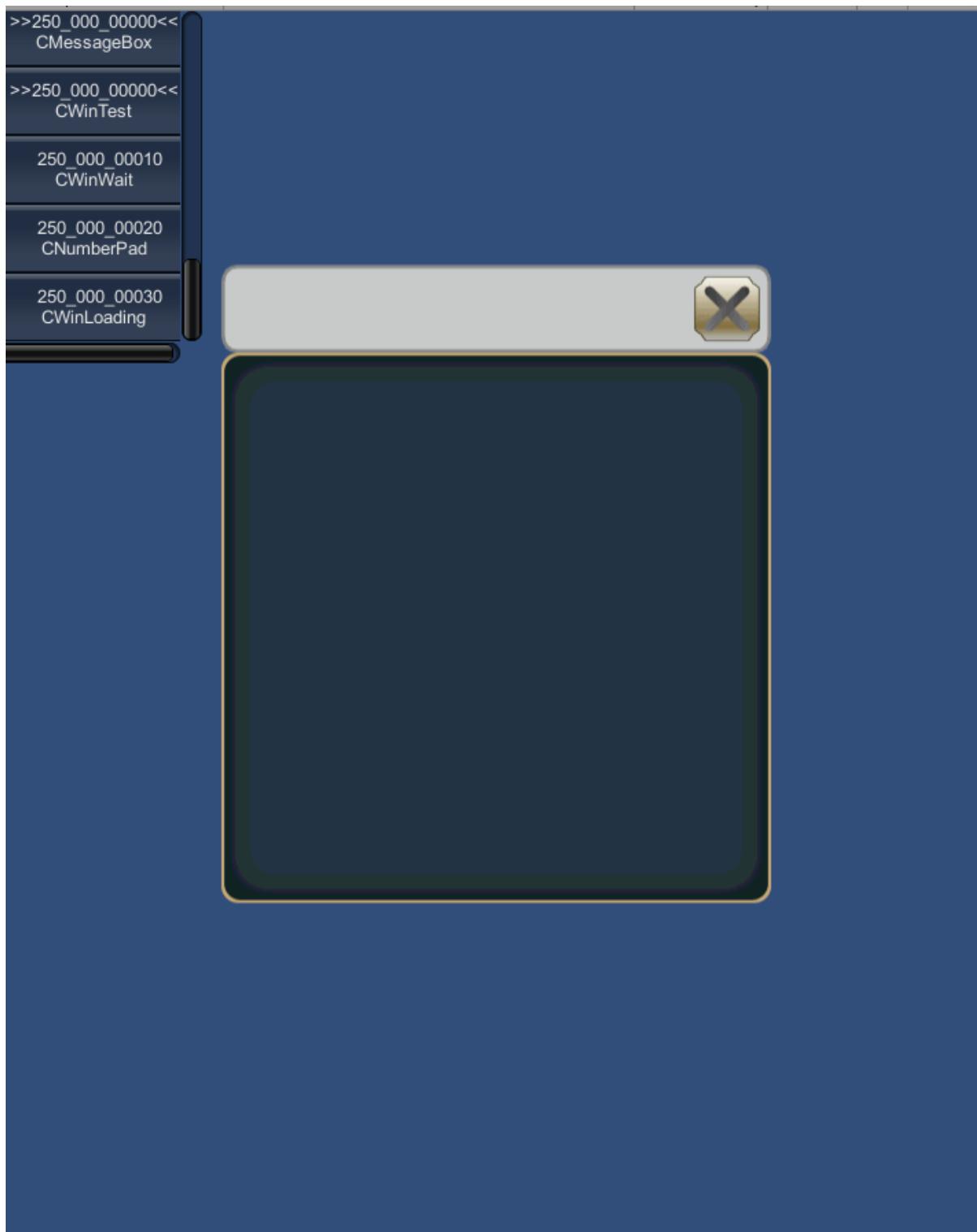
```
WINDOW(250_000_00000) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_ANCHOR_CENTER;
    SIZE = 400, 400;
};
```

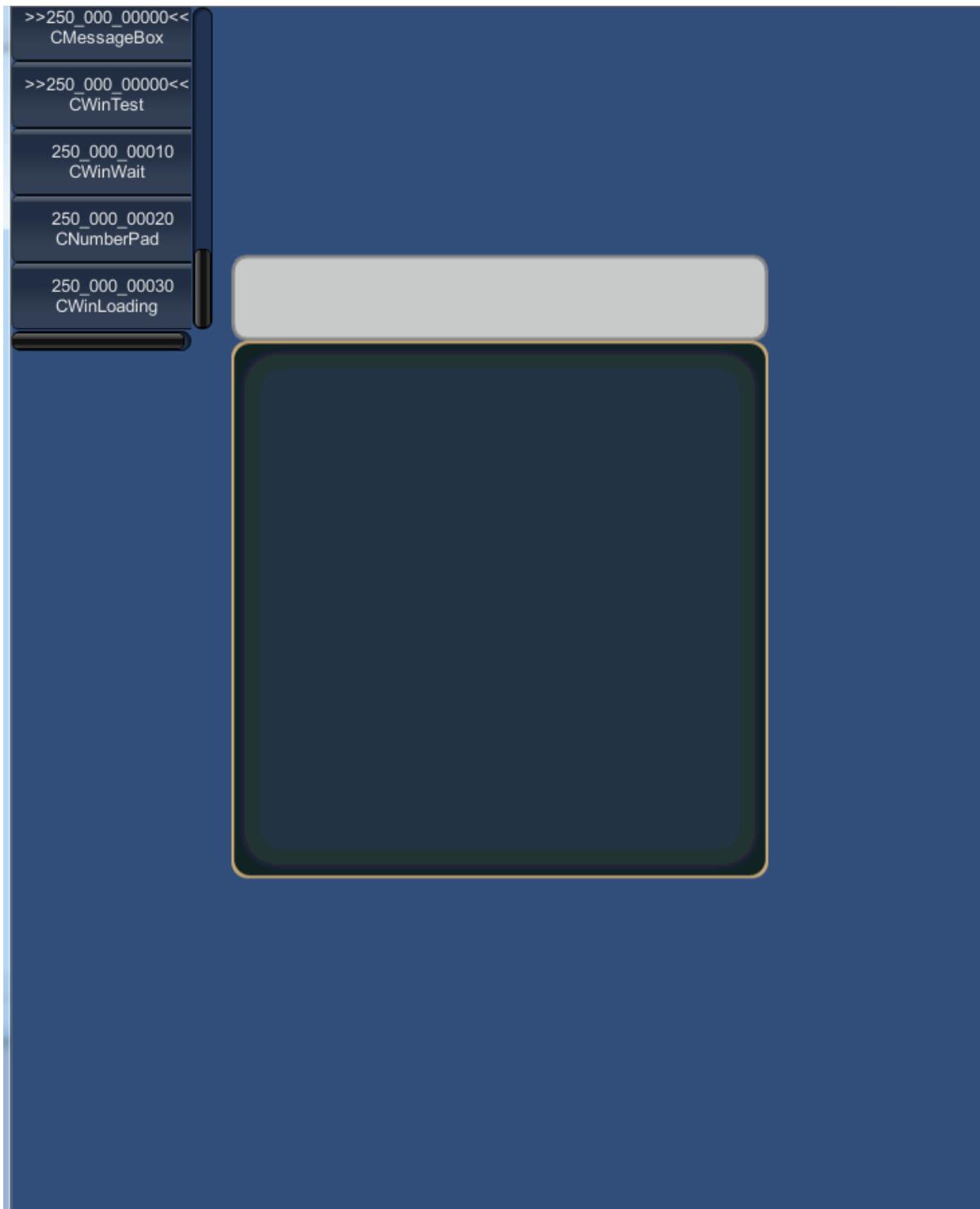
The window script which deleted all default indication

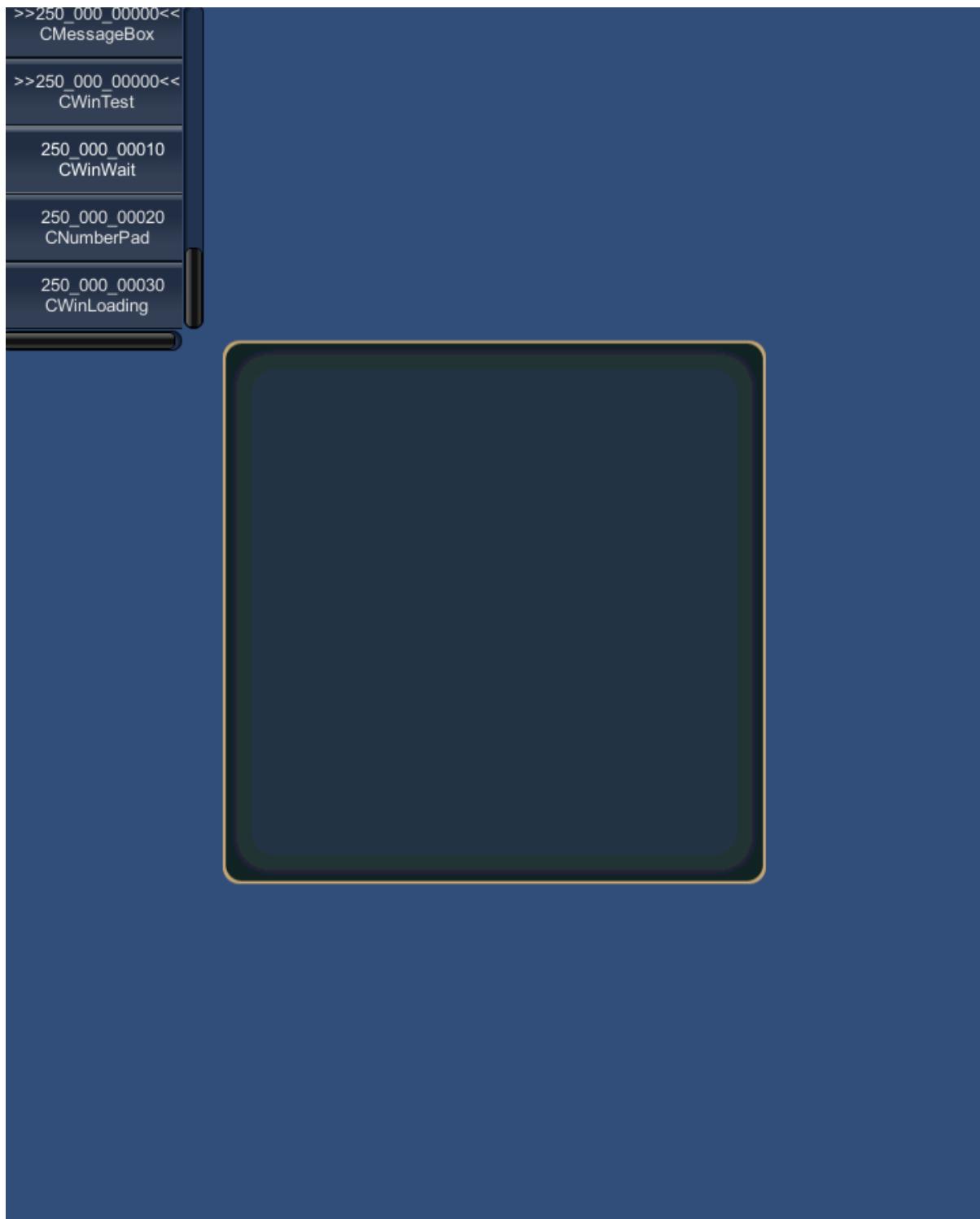
The window script which deleted all default indication

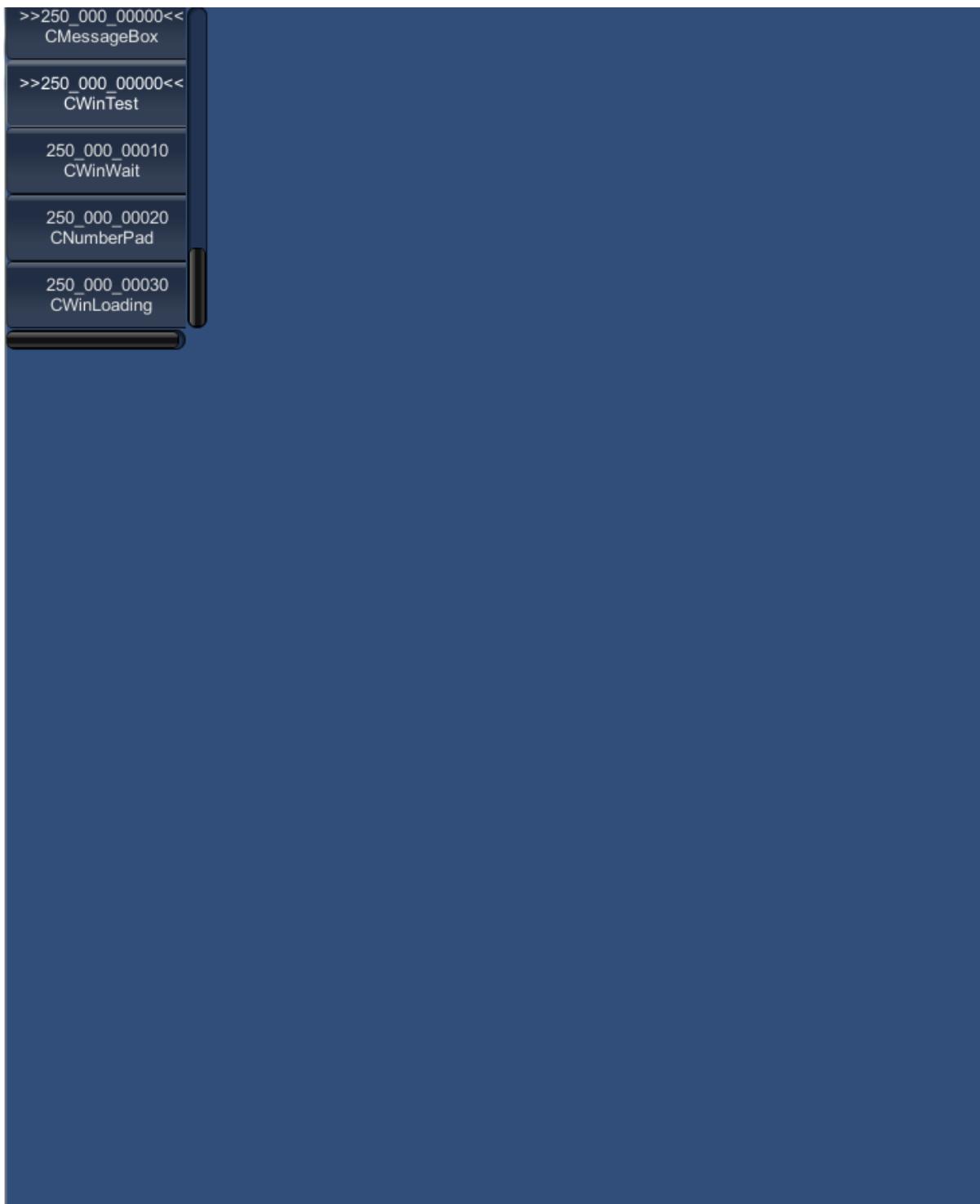
```
WINDOW(250_000_00000) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_NOFRAME|WINDOW_STYLE_ANCHOR_CENTER;
    SIZE = 400, 400;
};
```

The window script which deleted all default indication.









About the drug movement of the window

You can allow the movement of the window by WINDOW_STYLE_DRAG adding to STYLE.

When you drag the title bar, the frame, a user-defined control, you can move the window. But the window movement is not possible when a style of the control has WIN_CTRL_STYLE_DRAG. Please try the following examples.

```
WINDOW(250_000_00000) {
    STYLE = WINDOW_STYLE_ANCHOR_CENTER|WINDOW_STYLE_DRAG|WINDOW_STYLE_NOFRAME;
    POSITION = 0,0;
    CLOSE_POSITION = 400,0;
    SIZE = 400,400;
};

FRAME(Test) {
    STYLE = WIN_CTRL_STYLE_DRAG;
    SIZE = 400,400;
};
```

In this example, when you drag a frame, the frame is copied, it will move. On the other hand, you can move the window when you drag the title bar.

9.3.2 Property

RESOURCE = MulID

If you run a [KsTools]→[Export Window Resource], this system outputs it as an asset bundle.

If you set the same *MulID* to some window, these window data are included in an asset bundle together.

```
RESOURCE = 000_014_00000;
```

In this example, it outputs a 000_014_00000.unity3d as an asset bundle.

RESOURCE = output path

It outputs the asset data to the specified path.

If you set the same path to some window, these window data are included in an asset data together.

```
RESOURCE = "Assets/KsSoft/Resources/windows";
```

In this example It outputs the asset of “Assets/KsSoft/Resources/windows.asset.”

PATH = window base class output path

It sets a path to output the window base class.

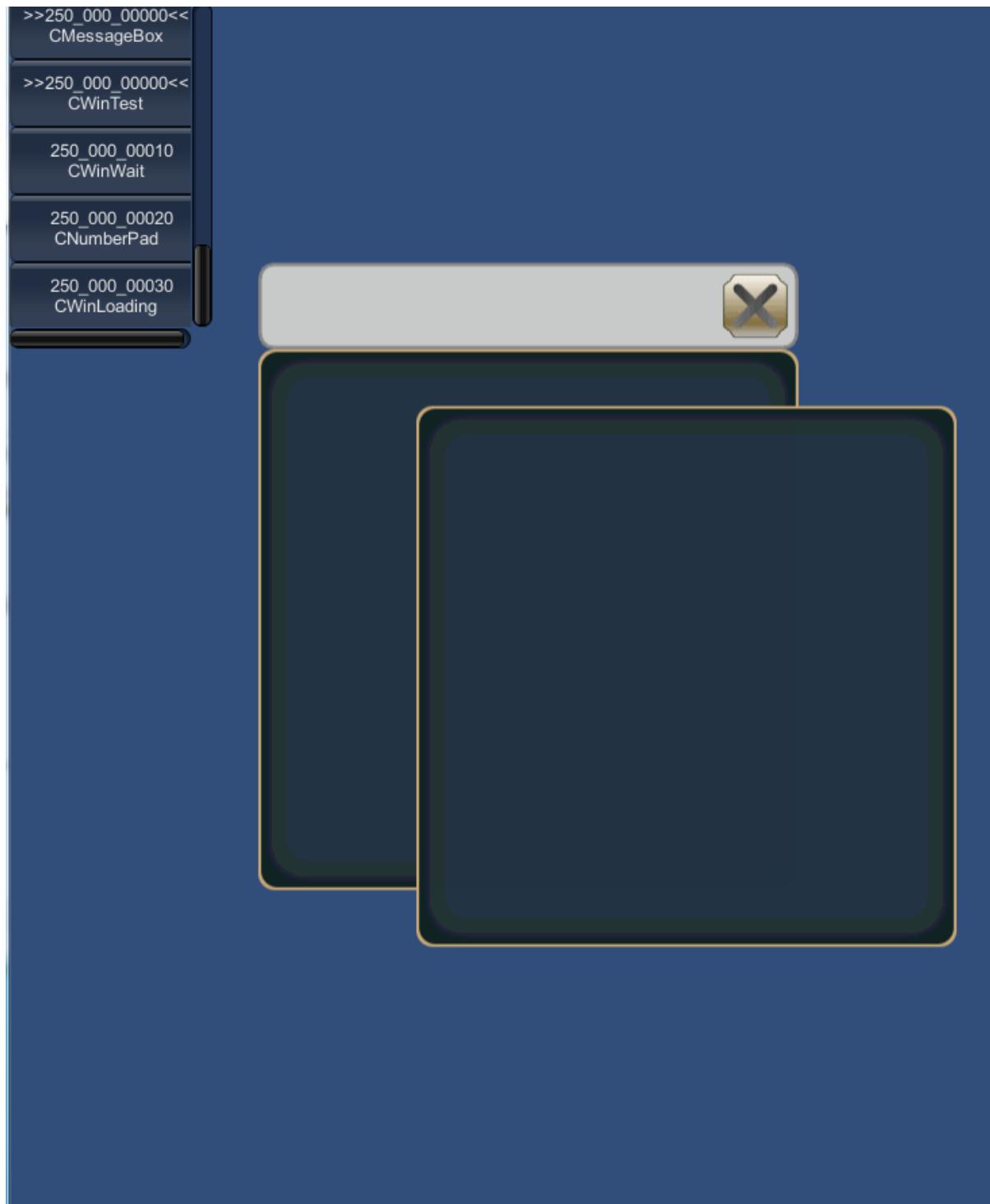
The current path is directly under the Unity project.

```
PATH = "../../../../client/Assets/Script/TestWindow.cs";
```

TEX_ID = Texture ID

You set the default texture ID.

This value is applied as the default texture ID of each control.



```
TEX_ID = 010_000_00010;
```

TEX_ID 0~7 = Texture ID

You set the default texture ID.

If you omit the texture ID of the control, this value is used.

In addition, you use this property in the case you want to select the texture for the frame and title bar.

CAPTION = Caption ID

You set the window's caption string.

If **WINDOW_STYLE_NOTITLEBAR** is set to **STYLE**, this value is ignored.

```
CAPTION = 020_000_00010;
```

POSITION = X, Y

To determine the window display position. Display position changes by an anchor that you specify in **STYLE**.

SIZE = width,height

Set the window size

By using the **RELATIVE_SIZE**, it can be set in a relative from the screen size.

```
SIZE = RELATIVE_SIZE(-100),RELATIVE_SIZE(-200);
```

In this example, if the screen size is 640x960, it makes the window of 540x760.

SCREEN = left top X,left Top Y,width,height

Set the position and size of the screen.

A window size and position are able to determine along the screen.

A window position and **RELATIVE_SIZE** are calculated on the basis of the screen.

If you do not set the screen, the screen is set to full screen.

```
SCREEN = 0,0,RELATIVE_SIZE(0),RELATIVE_SIZE(0); //全画面と同義
```

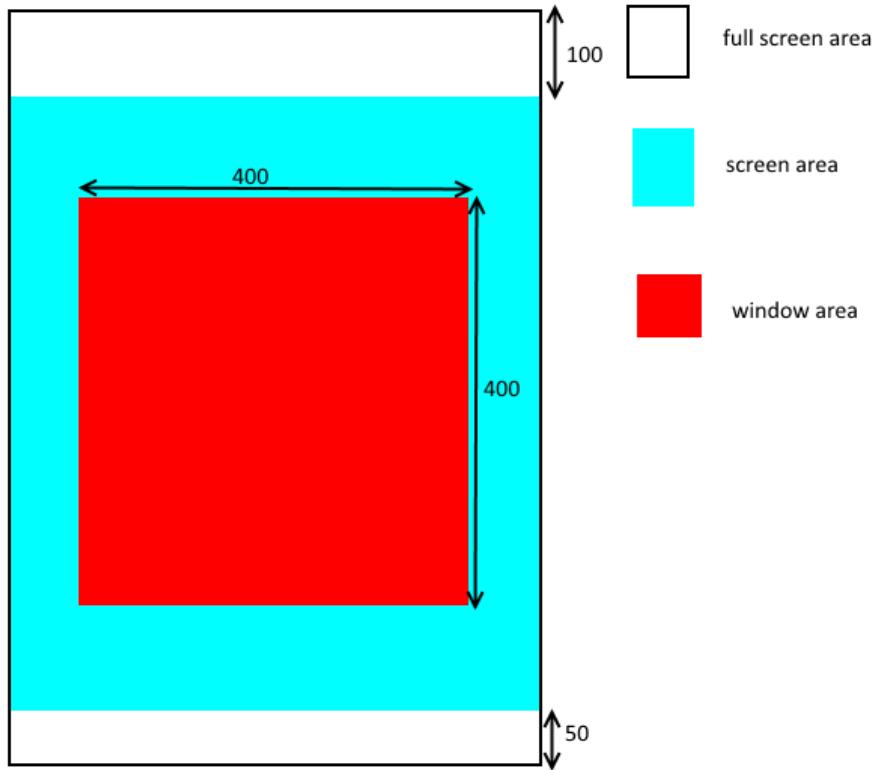
By using the **RELATIVE_SIZE**, it can be set in a relative from the screen size.

```
STYLE = WINDOW_STYLE_ANCHOR_CENTER;
SCREEN = 0,-100,RELATIVE_SIZE(0),RELATIVE_SIZE(-150);
SIZE = 400,400;
```

In this example, It places the window as shown in the figure below.

PRIORITY = display priority

Set the display priority of window. The higher the value, display priority is higher. Please be careful to differences with **TEXTURE_ZOFFSET**.



```
PRIORITY = 32;
```

TEXTURE_ZOFFSET = Texture ID,Z offset

If controls of window use the same texture, this window system renders with single mesh.

Therefore, the priority between the control of window means the display priority between the control which assigned the same texture.

To change the display priority between each texture, you can use the TEXTURE_ZOFFSET. You can change the display priority of the mesh by this property.

The smaller this value, display priority is higher.

```
TEXTURE_ZOFFSET = 014_000_00010, -1; //手前に表示するように変更
```

STYLE = style flag 0|style flag 1|..|style flag n

Display position anchor flag	Description
WINDOW_STYLE_ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_CENTER
WINDOW_STYLE_ANCHOR_LEFTTOP	Set the anchor position in the upper left
WINDOW_STYLE_ANCHOR_LEFT	Set the anchor position to the left. And vertical centering
WINDOW_STYLE_ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge
WINDOW_STYLE_ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral.
WINDOW_STYLE_ANCHOR_CENTER	Set the anchor position in the center of the window
WINDOW_STYLE_ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral.
WINDOW_STYLE_ANCHOR_RIGHTTOP	Set the anchor position in the upper right
WINDOW_STYLE_ANCHOR_RIGHT	Set the anchor position to the right And vertical centering
WINDOW_STYLE_ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge

Window display priority flag	Description
WINDOW_STYLE_TOP	Set the display priority of the window to maximum. This is a lower priority than POPUP and TOPMOST. If the same priority flag has been specified, display priority is determined by the PRIORITY property.
WINDOW_STYLE_POPUP	Set the display priority of the window to maximum. It is priority than TOP, and a lower priority than TOPMOST. If the same priority flag has been specified, display priority is determined by the PRIORITY property. if you touch the screen other than this window, this window is closed.
WINDOW_STYLE_TOPMOST	Set the display priority of the window to maximum. This is a higher priority than TOP and POPUP. If the same priority flag has been specified, display priority is determined by the PRIORITY property.
WINDOW_STYLE_NOECLIPSE	Even if this window is opened, you specify it when you do not want to darken the other window.(valid only when you have specified the TOP / POPUP)

Window function control flag	Description
WINDOW_STYLE_NOCLOSE	Do not place the close button. When NOTITLEBAR is attached, close button is not placed
WINDOW_STYLE_NOMINIMIZATION	Unimplemented
WINDOW_STYLE_NOHELP	Unimplemented
WINDOW_STYLE_NOTITLEBAR	Do not display the title bar.
WINDOW_STYLE_NOFRAME	Do not display the frame.
WINDOW_STYLE_DISABLE	Stop the function, can't accept input.
WINDOW_STYLE_DRAG	Allow the window move with dragging.
WINDOW_STYLE_NOACTIVE	It does not become active.
WINDOW_STYLE_HIDE	Turn off the window display.
WINDOW_STYLE_NOBRINGTOTOP	The display priority of a window doesn't allow to raise automatically when it becomes active.
WINDOW_STYLE_OPENBOTTOM	Open the window on the back.

STYLE = WINDOW_STYLE_NOTITLEBAR WINDOW_STYLE_NOFRAME;

Control:

Common Specification:

10.1 An arrangement of the control

10.1.1 Anchor:

It is possible to set the position of the origin of the window for each control.

You can specify the position of the control by using the relative position from the origin.

Thus, even if the screen size is changed, you can get a flexible layout not to collapse it.

We call it the anchor in this window system.

You can select nine kind anchor as shown in the figure below.

10.1.2 The base position

You can set the center position of the control to the individual.

Setting of the center position can be set from a location of nine kind.

10.1.3 The relationship between the anchor and the base position

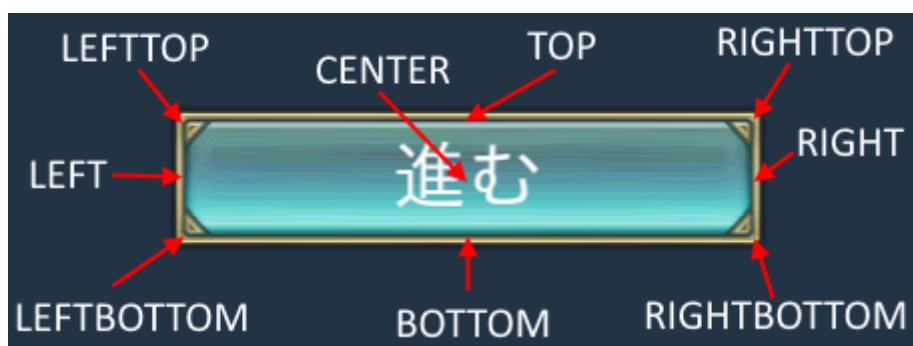
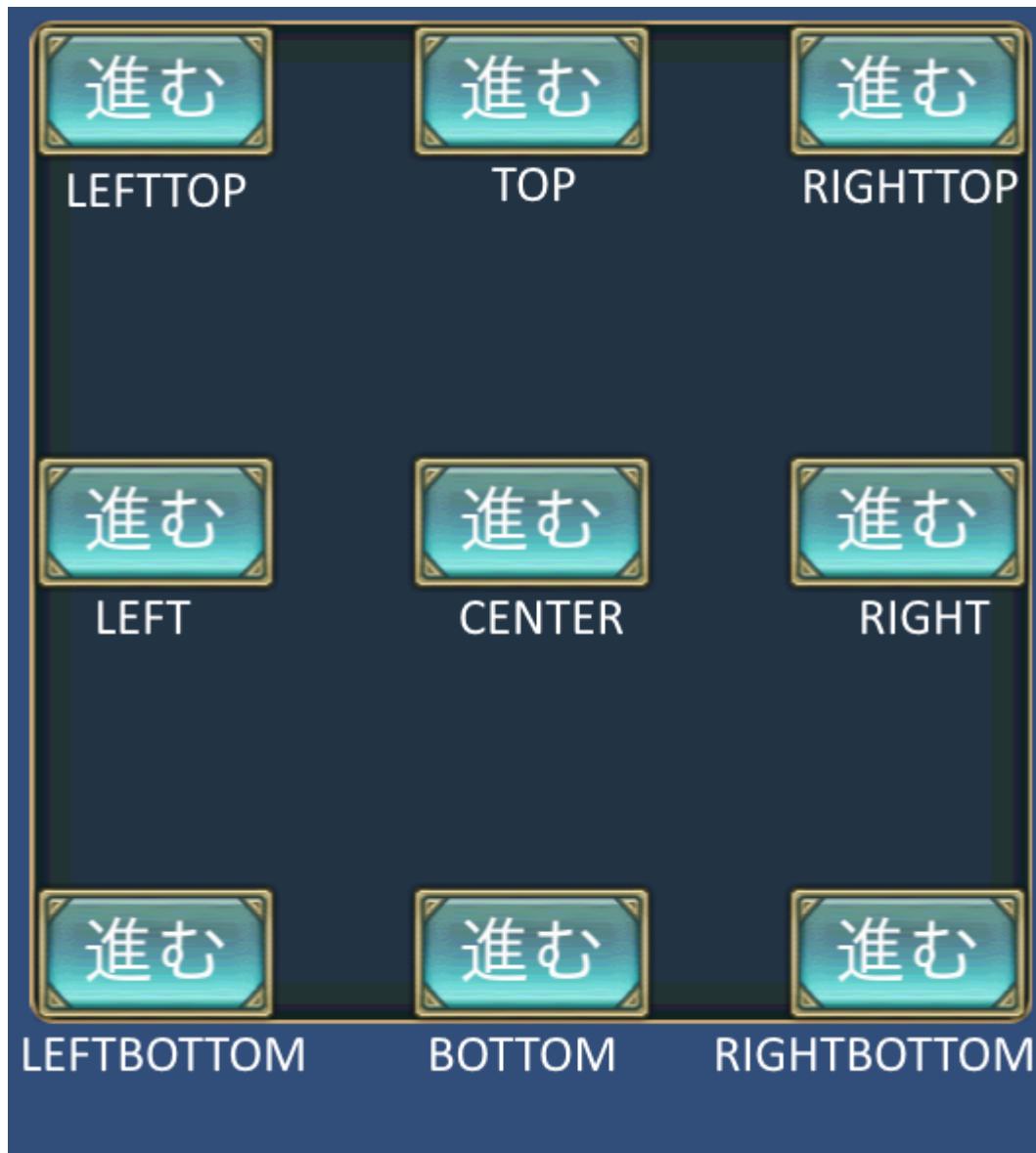
Depending on the anchor, the default value of the base position decide.

For example, when the anchor is set in the upper-left base position, it will be set to the upper left corner.

If you want to set the base position other than the default value, you need to explicitly set the value.

Enumerate the default value of the base position corresponding to the anchor value at below.

Anchor	Base position
ANCHOR_DEFAULT	BASE_DEFAULT
ANCHOR_LEFTTOP	BASE_LEFTTOP
ANCHOR_LEFT	BASE_LEFT
ANCHOR_LEFTBOTTOM	BASE_LEFTBOTTOM
ANCHOR_TOP	BASE_TOP
ANCHOR_CENTER	BASE_CENTER
ANCHOR_BOTTOM	BASE_BOTTOM
ANCHOR_RIGHTTOP	BASE_RIGHTTOP
ANCHOR_RIGHT	BASE_RIGHT
ANCHOR_RIGHTBOTTOM	BASE_RIGHTBOTTOM



10.2 The control priority

A window is rendered using a mesh .

A mesh is generated in corresponding to one texture.

Thus, the draw call does not increase even if the number of window control increase.

However , there is a need to be careful about the priority between the control .

Priority between the control to be rendered in a single draw call can be flexibly changed .

On the other hand , priority between different texture (= different mesh) is dependent on the priority between the texture priority

First , it is sorted by the textures , it is then sorted in the control priority within the same texture .

For example, the following window is composed by four meshes (textures) .



This window is constructed from two font meshes and two texture meshes.

- fn40
- fn30
- 100_000_00001
- 100_010_00000

10.2.1 If the same texture ID is specified

It can be set by the *PRIORITY* of the control .

A control with a large *PRIORITY* value is displayed in the front .

```
BUTTON(Button) {
    ID = 001_000_00010;
    POSITION = 100,100;
    SIZE = 64;
    PRIORITY = 5;
};
```

If the priority is the same

If the priority is the same , it is automatically sorted according to the type of controls .

The control which list order is high is rendered to the front.

- SCROLLBAR
- LISTBOX
- LISTBOXEX
- CONTAINER
- TEXT
- RICHTEXT
- LOG
- LOGTEXT
- BUTTON
- CHECKBOX
- RADIO
- HELPBUTTON
- EDITBOX
- TEXTBOX
- METER
- ICON
- CARD
- TEXTURE
- RENDER
- RENDERICON
- RECASTICON
- LINE
- LABEL
- FRAME
- BAR

10.2.2 If the texture ID is different

If the texture ID is different , The texture's z offset is priority than the priority of the control.

It is possible to set the Z offset of texture by *TEXTURE_ZOFFSET* of window properties .

The mesh that has larger value is displayed at the back.

```
WINDOW(100_100_00010) {
    TEX_ID = DEFAULT_TEXTURE;
    STYLE = WINDOW_STYLE_POPUP|WINDOW_STYLE_NOFRAME|WINDOW_STYLE_NOTITLEBAR;
    TEXTURE_ZOFFSET = 100_100_00000,-1;
    TEXTURE_ZOFFSET = 100_010_00000,1;
    POSITION = 0,0;
    SCREEN = 0,0,RELATIVE_SIZE(0),RELATIVE_SIZE(-BASEY);
    SIZE = WIN_W,WIN_H;
};
```

In this example , Meshes are rendered in the following order.

- 100_100_00000 front
- DEFAULT_TEXTURE
- 100_010_00000 back

Note: If not explicitly specify the Z offset, this window system assigns a value as this texture mesh is rendered in front of the existing meshes. This order is difficult to explicitly control because it is handled automatically by the system. If your window has three or more texture, you are better to set the *TEXTURE_ZOFFSET*.

Note: You can change the rendering order of generated meshes for render target by setting this *TEXTURE_ZOFFSET*.

refer to the *RENDER* , *RENDERICON* .

10.3 The control size

10.3.1 The relative size :

You can set the control size in relative size of the window. Set as follows.

```
SIZE = RELATIVE_SIZE(-16),RELATIVE_SIZE(16);
```

In this example it is set as the following.

width = window width - 16

height = window height + 16

10.3.2 The size default value

The default value is set when the size of the control was omitted or specify 0 .

The default value is the size of the texture parts.

If you want to display by dot-by-dot , you set to 0 or don't set.

10.4 Control common properties

It is enumerated the common property of controls.

Even if you set the value to some property depending on the control type, it may meaningless.

For more information, please look at the description of each control .

10.4.1 ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

10.4.2 POSITION = X, Y

It decides the display position . Please refer to the *anchor* .

```
POSITION = 32,-64;
```

10.4.3 SIZE = width,height

It changes the display size. If it is omitted , or , you specify 0 ,it gets the size from the texture parts.

```
SIZE = 64,32; //64x32
SIZE = ,32; //Set the width of the texture part width
SIZE = 64; //Set the height of the texture part height
```

10.4.4 CONTENTS = { control define … }

It defines contents int the control.

```
CONTENTS = {
    BUTTON(A) {
        :
    };
    CHECKBOX(B) {
        :
    };
    LISTBOX(C) {
        :
    };
}
```

10.4.5 CONTENTS_SIZE = width, height

It defines the size of the content.

10.4.6 LINE_SPACE = line spacing pixel value

It defines the gaps between the content.

```
LINE_SPACE = 8; //Put an 8-dot space.
```

10.4.7 GROUP = Control ID,…

It defines the group in the control. It is also used when you want to associate a listbox and scrollbar.

```
GROUP = RADIO(Spring),RADIO(Summer),RADIO(Autumn),RADIO(Winter),001_002_30000;
```

10.4.8 PRIORITY = Priority value

It defines the priority value . The higher the value, it will appear on front.

```
PRIORITY = 3;
```

10.4.9 TEX_ID = Texture ID,Part ID

10.4.10 TEX_ID = Part ID

You set the texture ID and the part ID.

10.4.11 TEX_ID n = Texture ID,Part ID

10.4.12 TEX_ID n = Part ID

You set the texture ID and the part ID.

It can be specified in the range of n = [0..7].

If n = 0, it operates the same texture as the TEX_ID.

10.4.13 TEXTURE_OFFSET n = offset X,offset Y

It defines the display offset of the texture .

It can be specified in the range of n = [0..7].

10.4.14 TEXTURE_SIZE n = width,height

You set the texture ID and the part ID.

It can be specified in the range of n = [0..7].

when n = 0 is the same meaning as ** SIZE **.

```
TEXTURE_SIZE1 = 64,32; //64x32
TEXTURE_SIZE2 = ,32;   //Set the width of the texture part width
TEXTURE_SIZE3 = 64;    //Set the height of the texture part height
```

10.4.15 COLOR = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

10.4.16 COLOR n = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

It can be specified in the range of n = [0..7].

If n = 0, it operates the same texture as **COLOR** .

10.4.17 CAPTION = Caption ID

Set the ID of the string .

Please refer to [here](#) .

```
CAPTION = 010_000_00100;
```

10.4.18 CAPTION = “string”

Set the string.

```
CAPTION = "Hello world!";
```

10.4.19 CAPTION_COLOR = R,G,B,A

You can set the caption color .

Set a value in the range of 0~1.

10.4.20 CAPTION_OFFSET = X, Y

It adjust the caption position.

10.4.21 SE_ID = SE_ID

To set the ID of the sound played when pressed . 0 is set as the default. This default value means not to play a sound.

10.4.22 STYLE = Flag0|Flag1|..|Flagn

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

You set the caption position to the center.

Text anchor	Description
TEXT_CENTER	It aligns the text to the center.
TEXT_LEFT	It aligns the text to the left side.
TEXT_RIGHT	It aligns the text to the right side.

Change the font decoration of the caption .

You can change the font decoration of the caption .	Description
TEXT_NORMAL	No decoration
TEXT_BOLD	Bold type
TEXT_DENT	Dent type
TEXT_SHADOW	Shadow Type

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.

Detailed specifications of each control:

10.5 TEXT

C#: *CWinCtrlText*

It is a control for displaying text .

If you want to change the text color,insert line feeds,or change the font type, please use the *RICHTEXT*

```
TEXT(Control name) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.5.1 Example

```
WINDOW(255_000_00001) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,-100;
    TEX_ID = 100_000_00001;
    CAPTION = 000_000_00010;
    SIZE = 256,256;
};

TEXT(TEST) {
    ID = 000_002_00001;
    CAPTION = 000_000_00006;
    FONT_KIND="fn40";
    POSITION = 0,0;
};
```



10.5.2 Property

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

POSITION = X, Y

It determine the display position.

```
POSITION = 32,-64;
```

FONT_KIND = Font kind

Set the font kind

```
FONT_KIND = "fnt32";
```

CAPTION = Caption ID

Set the ID of the string .

Please refer to [here](#) .

```
CAPTION = 010_000_00100;
```

CAPTION = “string”

Set the string.

```
CAPTION = "Hello world!";
```

CAPTION_COLOR = R,G,B,A

You can set the caption color .

Set a value in the range of 0~1.

CAPTION_OFFSET = X, Y

It adjust the caption position.

STYLE = Flag0|Flag1|..|Flagn

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

You set the caption position to the center.

Text anchor	Description
TEXT_CENTER	It aligns the text to the center.
TEXT_LEFT	It aligns the text to the left side.
TEXT_RIGHT	It aligns the text to the right side.

Change the font decoration of the caption .

You can change the font decoration of the caption .	Description
TEXT_NORMAL	No decoration
TEXT_BOLD	Bold type
TEXT_DENT	Dent type
TEXT_SHADOW	Shadow Type

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.

10.6 RICHTEXT

C#: *CWinCtrlRichText*

It is a control for displaying rich text.

It is possible to change the text color , insert line feeds, or change the font type

```
RICHTEXT(Control name) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.6.1 Example

```
WINDOW(255_000_00001) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_ANCHOR_LEFT;
    POSITION = 0,-100;
    SIZE = 512,256;
};

RICHTEXT(Message) {
    ID = 00_000_00010;
    POSITION = 0,-64;
    CAPTION = "\a[3]TESTabc\f[fn16]smallFont\t123[100_000_00001,ATTR1,32,32,0]" +
               "\a[4]\w[001_002_00003,click here!]";
    CONTENTS_SIZE = 380, 64;
};
```



10.6.2 Property

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

SIZE = new line width

When the string is longer than new line width , it will be a new line automatically.

```
SIZE = 128; //128pixel
```

FONT_KIND = Font kind

Set the font kind

```
FONT_KIND = "fnt32";
```

CAPTION = Caption ID

Set the ID of the string .

Please refer to [here](#) .

```
CAPTION = 010_000_00100;
```

CAPTION = “string”

Set the string.

```
CAPTION = "Hello world!";
```

CAPTION_COLOR = R,G,B,A

You can set the caption color .

Set a value in the range of 0~1.

LINE_SPACE = line spacing pixel value

Set the line spacing.

```
LINE_SPACE = 8;           //Put an 8-dot space.
```

CAPTION_OFFSET = X, Y

It adjust the caption position.

STYLE = Flag0|Flag1|..|Flagn

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

You set the caption position to the center.

Text anchor	Description
TEXT_CENTER	It aligns the text to the center.
TEXT_LEFT	It aligns the text to the left side.
TEXT_RIGHT	It aligns the text to the right side.

Change the font decoration of the caption .

You can change the font decoration of the caption .	Description
TEXT_NORMAL	No decoration
TEXT_BOLD	Bold type
TEXT_DENT	Dent type
TEXT_SHADOW	Shadow Type

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.

Available control code in RichText

It is possible to apply a variety decoration by embedding the control code in the caption of a rich text.

For example,if you want to insert the texture in the middle of a sentence , you set a string to the caption as follows.

```
"テクスチャテスト\t[001_001_01000,icon0,32,32,0]\n表示テスト"
```

Command	Description
\n or \r	Line break
\ \	\ Character
\f[font name]	Switch to the specified font.
\a[0] or \a[1]	Horizontal:Left side , Vertical:Top side
\a[2]	Horizontal:Left side , Vertical:Center
\a[3]	Horizontal:Left side , Vertical:Bottom side
\a[4]	Horizontal:Center , Vertical:Top side
\a[5]	Horizontal:Center , Vertical:Center
\a[6]	Horizontal:Center , Vertical:Bottom side
\a[7]	Horizontal:Right side , Vertical:Top side
\a[8]	Horizontal:Right side , Vertical:Center
\a[9]	Horizontal:Right side , Vertical:Bottom side
\c[0]	white color
\c[1]	Light blue color
\c[2]	Purple color
\c[3]	Blue color
\c[4]	Yellow color
\c[5]	Green color
\c[6]	Red color
\c[7]	Pink color
\c[8]	Orange color
\c[9]	Sky blue color
\c[10]	Black color
\C[00000000] ~ \C[FFFFFFFF]	set the color in hexadecimal (AARRGGBB)
\wUserID[WindowID,caption]	Window command of the set ID
\tUserID[TextureID,PartID,width,height,y offset]	Insert the texture <i>part</i> in the middle of a sentence.
\B or \b	Bold font
\N	Without decoration

If the anchor is changed by using the \a command(right,center,left), it put a line break into the sentence automatically.

Window command \w

The *callback* is called when you touch the set caption string.

The format is as follows.

\wUserID[WindowID,caption]

- UserID: Set the integer(this is option. The default is 0.).
- WindowID: set the MulID
- Caption: Set the string. The callback is called when you press this string.

```
CAPTION = "\a[4]\w123[001_002_00003,click here!];"
```

Texture command \t

Insert the texture *part* in the middle of a sentence.

Callback is called when you touch this texture part.

The format is as follows.

\tUserID[TextureID,PartID,width,height,y offset]

- UserID: Set the integer(this is option. The default is 0.).
- Texture ID: Set a MulID.
- width,height: Set the pixel size.
- Display Y offset(this is option).

```
CAPTION = "テクスチャテスト\t456[001_001_01000,icon0,32,32,0];"
```

10.7 LOG

C#: *CWinCtrlLog*

This is the control to display a chat log.

It has contents of the same number as the number of rows.

Unlike the list box , the maximum number of contents can be set. If it is added more than the maximum number , the contents is circulated and it is reused(The oldest log is overwritten).

LOGTEXT and other kind of controls can also be included as contents.

Note: The LOG's contents need to include only one *LOGTEXT*.

```
LOG(Control name) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.7.1 Example

```

LOG(Chat) {
    ID = 000_000_00030;
    STYLE = ANCHOR_LEFTBOTTOM;
    POSITION = 0,OFFSET_Y;
    SIZE = WINDOW_SIZE_FULL, 80-2;
    CONTENTS_SIZE = WINDOW_SIZE_FULL, 80-2;
    CONTENTS = {
        ICON(Chat) {
            :
        }
        TEXT(Name) {
            :
        }
        FRAME(Balloon) {
            :
        }
        LOGTEXT(Chat) {
            :
        }
    }

    COLOR = COLOR32(0,0,0,255);
    LINE_SPACE = 10;
    GROUP = SCROLLBAR(Chat);
};

SCROLLBAR(Chat) {
    ID = 000_001_00001;
    STYLE = ANCHOR_RIGHTTOP;
    COLOR = 1,1,1,0.5;
    POSITION = -2,-5;
    SIZE = 0,64-10;
};

```

10.7.2 Property

Default value

```

COLOR = 1,1,1,1;
CAPTION_COLOR = 1,1,1,1;

```

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position.

```
POSITION = 32,-64;
```

SIZE = width,height

Set the display area of the list box.

Control protruding from the display area is clipped .

```
SIZE = 400,400;           //display size
CONTENTS_SIZE = 800,80;    //one content size
```

CONTENTS_SIZE = width, height

Determine the listbox size of the line. A listbox which arranged in the vertical direction, scrolls in the vertical direction.

However ,if the width of the content size is larger than the display area, the listbox scrolls to the left or right.

```
// only vertical direction to scroll
SIZE = 400,400;           //display size
CONTENTS_SIZE = 400,40;     //virtual screen size

// Scroll to the left and right direction
SIZE = 400,400;           //display size
CONTENTS_SIZE = 800,40;     //virtual screen size
```

CONTENTS = { control define ... }

Enumerate the controls for defining a single line in the LOG.

Position of the controls place with a relative from the position of the LOG.

It is possible to include any type of control .

Note: The LOG's contents need to include only one *LOGTEXT*.

```
CONTENTS = {
ICON(Chat) {
:
}
TEXT(Name) {
:
}
FRAME(Balloon) {
:
}
LOGTEXT(Chat) {
:
}
```

How to access to a particular control of the content list from C#, please refer to [here](#).

However, since the content list is in the circulation list , it is dangerous to access directly. It should be set when you *add a log* .

LINE_SPACE = line spacing pixel value

It defines the gaps between the content.

```
LINE_SPACE = 8;           //Put an 8-dot space.
```

GROUP = SCROLLBAR Control ID,...

If you set scroll controls to the GROUP property,it display scroll bars in conjunction with the log.

It should be careful not to set to CONTENTS.

It is also possible to assign multiple scrollbars.

GROUP = SCROLLBAR(Horizontal), SCROLLBAR(Vertical);

COLOR = R,G,B,A

It is possible to set the color values . By changing color , it affects all of the controls that are included

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

STYLE = Flag0|Flag1|..|Flagn

Configurable unique styles in the list box are like the following .

Listbox control flag	Description
WIN_CTRL_STYLE_NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.
LISTBOX_LOCK	Suppress that scroll the listbox.

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.
WIN_CTRL_STYLE_NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.
LISTBOX_LOCK	Suppress that scroll the listbox.

10.8 LOGTEXT

C#: *CWinCtrlLogText*

This is a control for having the sentence of the *chat log*.

Note: The *LOG's contents* always requires a LOGTEXT.

```
LOGTEXT(Control name) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.8.1 Example

```
LOG(Chat) {
    ID = 000_000_00030;
    STYLE = ANCHOR_LEFTBOTTOM;
    POSITION = 0,OFFSET_Y;
    SIZE = WINDOW_SIZE_FULL, 80-2;
    CONTENTS_SIZE = WINDOW_SIZE_FULL, 80-2;
    CONTENTS = ICON(Chat),TEXT(Name),FRAME(Balloon),LOGTEXT(Chat);
    COLOR = COLOR32(0,0,0,255);
    LINE_SPACE = 10;
    GROUP = SCROLLBAR(Chat);
};

SCROLLBAR(Chat) {
    ID = 000_001_00001;
    STYLE = ANCHOR_RIGHTTOP;
    COLOR = 1,1,1,0.5;
    POSITION = -2,-5;
    SIZE = 0,64-10;
};

LOGTEXT(Chat) {
    ID = 000_001_00100;
    STYLE = NOHIT;
    POSITION = 105, -44;
    CAPTION_COLOR = 0,0,0,1;
    FONT_KIND = "fn24";
    CAPTION = 000_000_00010;
    CONTENTS_SIZE = WINDOW_SIZE_FULL -235, 25;
    SIZE = WINDOW_SIZE_FULL-135,55;
};
```

10.8.2 Property

Default value

```
COLOR = 1,1,1,1;
CAPTION_COLOR = 1,1,1,1;
```

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position.

```
POSITION = 32,-64;
```

SIZE = width,height

Set the display area of the list box.

Control protruding from the display area is clipped .

```
SIZE = 400,400;           //display size
CONTENTS_SIZE = 800,80;    //one contents size
```

FONT_KIND = Font kind

Set the font kind

```
FONT_KIND = "fnt32";
```

CAPTION = Caption ID

Set the ID of the string .

Please refer to [here](#) .

```
CAPTION = 010_000_00100;
```

CAPTION = “string”

Set the string.

```
CAPTION = "Hello world!";
```

CAPTION_COLOR = R,G,B,A

You can set the caption color .

Set a value in the range of 0~1.

CAPTION_OFFSET = X, Y

It adjust the caption position.

STYLE = Flag0|Flag1|..|Flagn

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

You set the caption position to the center.

Text anchor	Description
TEXT_CENTER	It aligns the text to the center.
TEXT_LEFT	It aligns the text to the left side.
TEXT_RIGHT	It aligns the text to the right side.

Change the font decoration of the caption .

You can change the font decoration of the caption .	Description
TEXT_NORMAL	No decoration
TEXT_BOLD	Bold type
TEXT_DENT	Dent type
TEXT_SHADOW	Shadow Type

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.

10.9 EDITBOX

C#: *CWinCtrlEditbox*

It is a text editable control.

You can set the maximum rows and the maximum string size.

```
EDITBOX(Control name)
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.9.1 Example

Examples of line input edit box

```
EDITBOX(LoginID) {
    ID = 001_000_00010;
    STYLE = ANCHOR_BOTTOM;
    SIZE = 300;
    EDIT = 255,0;           //255 characters, one line
    POSITION = 0,$y;
    FONT_KIND = "fn24";
};
```

Examples of multi-line input edit box(10 lines in this example).

```
EDITBOX(Sentence) {
    ID = 000_000_00010;
    STYLE = WIN_CTRL_STYLE_TEXT_LEFT;
    CAPTION_COLOR = COLOR32(50,50,50,255);
```

```

FONT_KIND="fn16";
EDIT = 255,10;           //255 characters,10 lines
TEX_ID = 0,"EDBT0";
POSITION = $x,$y;
SIZE = $w,$h;
} ;

```

10.9.2 Property

Default value

```

TEX_ID = "TXXFD?";
COLOR = 1,1,1,1;
CAPTION_COLOR = 1,1,1,1;

```

Part ID rule

It replaces the fifth character of the part ID to 0/1.

- XXXX0: off state
- XXXX1: on state

```
TEX_ID = "TXXFD?";
```

If parts ID has been set in this way, you need to prepare parts, such as the following.

- TXXFD0: Focus is in the state of OFF
- TXXFD1: Focus is in the state of ON

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

EDIT = input character maximum number, input possible number of lines

Set the input character maximum number and input possible number of lines.

if a edit box is input possible number of lines that is 0 or 1, it is the line input edit box.

POSITION = X, Y

It determine the display position . By the setting of STYLE, you can change the criteria for display position.

```
POSITION = 32,-64;
```


CAPTION_COLOR = R,G,B,A

You can set the caption color .

it becomes the color of the input string in EDITBOX.

Set a value in the range of 0~1.

CAPTION_OFFSET = X, Y

It adjust the caption position.

**** STYLE ** = style 0 | style 1 | .. | style n**

set the editbox style.

The editbox style	Description
EDIT_BLIND	Mask the input characters
EDIT_ALL	The default is possible that all of the character input.
EDIT_TYPE_ASCIICAPABLE	ASCII array keyboard
EDIT_TYPE_URL	URL input keyboard
EDIT_TYPE_NUMBERANDPUNCTUATION	Number and punctuation keyboard
Numberpad keyboard	PIN keyboard
EDIT_TYPE_PHONEPAD	Phone pad keyboard
EDIT_TYPE_NAMEPHONEPAD	name phone pad keyboard
EDIT_TYPE_EMAILADDRESS	email address keyboard

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

You set the caption position to the center.

Text anchor	Description
TEXT_CENTER	It aligns the text to the center.
TEXT_LEFT	It aligns the text to the left side.
TEXT_RIGHT	It aligns the text to the right side.

Change the font decoration of the caption .

You can change the font decoration of the caption .	Description
TEXT_NORMAL	No decoration
TEXT_BOLD	Bold type
TEXT_DENT	Dent type
TEXT_SHADOW	Shadow Type

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.
WIN_CTRL_STYLE_NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.

10.10 TEXTBOX

C#: *CWinCtrlTextbox*

This is a edit box,but it can't be edited.

If the caption does not fit to the control size , it is possible to scroll.

```
TEXTBOX(Control name) {
    Property 1;
    Property 2;
    :
    :
}
```

```
Property n
};
```

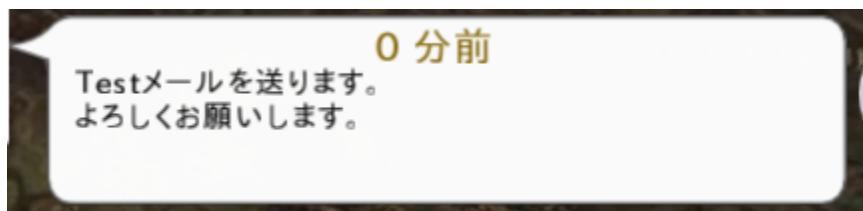
10.10.1 Example

Examples of one line of text box. (It becomes almost the same behavior as the TEXT.)

```
TEXTBOX(Sentence) {
    ID = 001_000_00010;
    STYLE = ANCHOR_BOTTOM;
    SIZE = 300;
    EDIT = 255,0;           //255 characters,one line
    POSITION = 0,$y;
    FONT_KIND = "fn24";
};
```

Multiline example of the text box (10 lines text box in this example.).

```
TEXTBOX(Sentence) {
    ID = 000_000_00010;
    STYLE = WIN_CTRL_STYLE_TEXT_LEFT;
    CAPTION_COLOR = COLOR32(50,50,50,255);
    FONT_KIND="fn16";
    EDIT = 255,10;          //255 characters,10 lines
    TEX_ID = 0,"EDBT0";
    POSITION = $x,$y;
    SIZE = $w,$h;
};
```



10.10.2 Property

```
TEX_ID = "TXXFD?";
COLOR = 1,1,1,1;
CAPTION_COLOR = 1,1,1,1;
```

Part ID rule

It replaces the fifth character of the part ID to 0/1.

- XXXX0: off state
- XXXX1: on state

```
TEX_ID = "TXXFD?";
```

If parts ID has been set in this way, you need to prepare parts, such as the following.

- TXXFD0: Focus is in the state of OFF
- TXXFD1: Focus is in the state of ON

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

EDIT = caption character maximum number, caption possible number of lines

Set the caption character maximum number and caption possible number of lines.

POSITION = X, Y

It determine the display position . By the setting of STYLE, you can change the criteria for display position.

```
POSITION = 32,-64;
```

SIZE = width,height

It changes the display size. If it is omitted , or , you specify 0 ,it gets the size from the texture parts.

```
SIZE = 64,32; //64x32
SIZE = ,32; //Set the width of the texture part width
SIZE = 64; //Set the height of the texture part height
```

TEX_ID = Texture ID,Part ID**TEX_ID = Part ID**

You set the texture ID and the part ID.

COLOR = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

CAPTION = Caption ID

Set the ID of the string .

Please refer to [here](#) .

```
CAPTION = 010_000_00100;
```


Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

You set the caption position to the center.

Text anchor	Description
TEXT_CENTER	It aligns the text to the center.
TEXT_LEFT	It aligns the text to the left side.
TEXT_RIGHT	It aligns the text to the right side.

Change the font decoration of the caption .

You can change the font decoration of the caption .	Description
TEXT_NORMAL	No decoration
TEXT_BOLD	Bold type
TEXT_DENT	Dent type
TEXT_SHADOW	Shadow Type

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.
WIN_CTRL_STYLE_NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.

10.11 BUTTON

C#: *CWinCtrlButton*

This is a button control .

It is possible to display some of the badge and the caption.

It can have seven badges at the maximum.

```
BUTTON(Control Name) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.11.1 Example

```
WINDOW(255_000_00001) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,-100;
    TEX_ID = 100_000_00001;
    CAPTION = 000_000_00010;
    SIZE = 512,256;
};

BUTTON(Button) {
    ID = 000_002_00001;
    CAPTION = 000_000_00006;
    STYLE = ANCHOR_CENTER;
    POSITION = 0,0;
    FONT_KIND= "fn40";
    TEX_ID1 = "bd01";
    TEXTURE_OFFSET1 = 0,10;
    SIZE = 256;
};
```



10.11.2 Property

Default value

```
TEX_ID = "BTNO?";  
COLOR = 1,1,1,1;  
CAPTION_COLOR = 1,1,1,1;  
SE_ID = clickSE;
```

Part ID rule

It replaces the fifth character of the part ID to 0/1.

- XXXX0: off state
- XXXX1: on state

```
TEX_ID = "BTNO?";
```

If parts ID has been set in this way, you need to prepare parts, such as the following.

- BNT00: off state
- BNT01: on state

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position . By the setting of STYLE, you can change the criteria for display position.

```
POSITION = 32, -64;
```

SIZE = width,height

It changes the display size. If it is omitted , or , you specify 0 ,it gets the size from the texture parts.

```
SIZE = 64,32; //64x32
SIZE = ,32;   //Set the width of the texture part width
SIZE = 64;     //Set the height of the texture part height
```

TEX_ID = Texture ID,Part ID

TEX_ID = Part ID

You set the texture ID and the part ID.

COLOR = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

CAPTION = Caption ID

Set the ID of the string .

Please refer to [here](#) .

CAPTION = “string”

Set the string.

```
CAPTION = "Hello world!";
```

CAPTION_COLOR = R,G,B,A

You can set the caption color .

Set a value in the range of 0~1.

CAPTION_OFFSET = X, Y

It adjust the caption position.

TEX_ID n = Texture ID,Part ID

TEX_ID n = Part ID

Set the texture ID and the part ID of the badge.

It can be set in the range of n = [1..7].

TEXTURE_OFFSET n = offset X,offset Y

Set the display offset of the badge.

It can be set in the range of n = [1..7].

TEXTURE_SIZE n = width,height

Set the texture ID and the part ID of the badge.

It can be set in the range of n = [1..7].

```
TEXTURE_SIZE1 = 64,32; //64x32
TEXTURE_SIZE2 = ,32;   //Set the width of the texture part width
TEXTURE_SIZE3 = 64;    //Set the height of the texture part height
```

COLOR n = R,G,B,A

Set the badge color .

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

It can be set in the range of n = [1..7].

SE_ID = SE_ID

Set the ID of the sound to sound when pressed . The default is clickSE. The sound will not sound when set to 0 .

STYLE = Flag0|Flag1|..|Flagn

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

You set the caption position to the center.

Text anchor	Description
TEXT_CENTER	It aligns the text to the center.
TEXT_LEFT	It aligns the text to the left side.
TEXT_RIGHT	It aligns the text to the right side.

Change the font decoration of the caption .

You can change the font decoration of the caption .	Description
TEXT_NORMAL	No decoration
TEXT_BOLD	Bold type
TEXT_DENT	Dent type
TEXT_SHADOW	Shadow Type

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.

10.12 RADIO

C#: *CWinCtrlRadio*

Radio buttons can be grouped radio buttons with each other. When one of the radio button is turned on, the other radio button automatically turns off.

It has the *CONTENTS* as *CHECKBOX* . So it is possible to behave as a tab button.

```
RADIO(Control name) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.12.1 Example

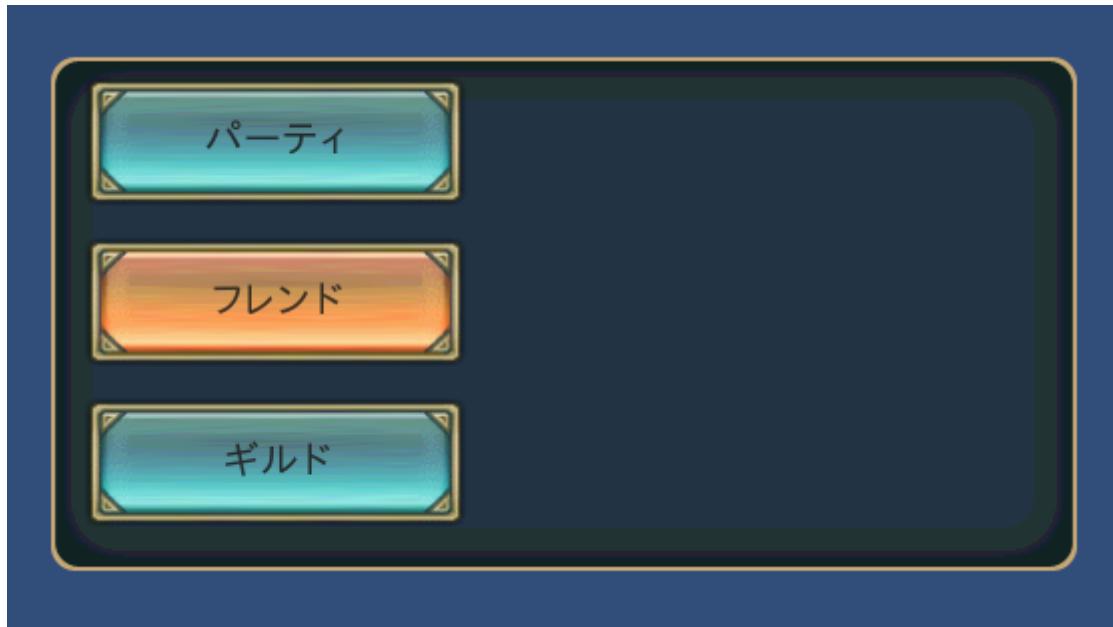
```
WINDOW(255_000_00001) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,-100;
    TEX_ID = 100_000_00001;
    CAPTION = 000_000_00010;
    SIZE = 512,256;
};

RADIO(Party) {
    ID = 000_000_00100;
    POSITION = 16,-10;
    TEX_ID=0, "BTNT?";
    CAPTION_COLOR=COLOR32(30,30,30,220);
    FONT_KIND="fn22";
    SIZE=44*4+16, 64;
    CAPTION = 000_000_00015;
    GROUP = RADIO(Party),RADIO(Friend),RADIO(Guild);
};

RADIO(Friend) {
    ID = 000_000_00110;
    POSITION = 16,-90;
    TEX_ID=0, "BTNT?";
    CAPTION_COLOR=COLOR32(30,30,30,220);
    FONT_KIND="fn22";
    SIZE=44*4+16, 64;
```

```
CAPTION = 010_000_00020;
GROUP = RADIO(Party),RADIO(Friend),RADIO(Guild);
};

RADIO(Guild) {
    ID = 000_000_00120;
    POSITION = 16,-170;
    TEX_ID=0, "BTNT?";
    CAPTION_COLOR=COLOR32(30,30,30,220);
    FONT_KIND="fn22";
    SIZE=44*4+16,64;
    CAPTION = 010_000_00010;
    GROUP = RADIO(Party),RADIO(Friend),RADIO(Guild);
};
```



In this example , party button , friend button , the guild button is exclusively turned on.

10.12.2 Property

Default value

```
TEX_ID = "BTN0?";
COLOR = 1,1,1,1;
CAPTION_COLOR = 1,1,1,1;
```

Part ID rule

It replaces the fifth character of the part ID to 0/1.

- XXXX0: off state
- XXXX1: on state

```
TEX_ID = "BTN0?";
```

If parts ID has been set in this way, you need to prepare parts, such as the following.

- BNT00: off state
- BNT01: on state

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position . By the setting of STYLE, you can change the criteria for display position.

```
POSITION = 32,-64;
```

SIZE = width,height

It changes the display size. If it is omitted , or , you specify 0 ,it gets the size from the texture parts.

```
SIZE = 64,32; //64x32
SIZE = ,32; //Set the width of the texture part width
SIZE = 64; //Set the height of the texture part height
```

TEX_ID = Texture ID,Part ID**TEX_ID = Part ID**

You set the texture ID and the part ID.

COLOR = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

SE_ID = SE_ID

Set the ID of the sound to sound when pressed . The default is clickSE. The sound will not sound when set to 0 .

CAPTION = Caption ID

Set the ID of the string .

Please refer to [here](#) .

```
CAPTION = 010_000_00100;
```

CAPTION = "string"

Set the string.

```
CAPTION = "Hello world!";
```

CAPTION_COLOR = R,G,B,A

You can set the caption color .

Set a value in the range of 0~1.

CAPTION_OFFSET = X, Y

It adjust the caption position.

TEX_ID n = Texture ID,Part ID**TEX_ID n = Part ID**

Set the texture ID and the part ID of the badge.

It can be set in the range of n = [1..7].

TEXTURE_OFFSET n = offset X,offset Y

Set the display offset of the badge.

It can be set in the range of n = [1..7].

TEXTURE_SIZE n = width,height

Set the texture ID and the part ID of the badge.

It can be set in the range of n = [1..7].

```
TEXTURE_SIZE1 = 64,32; //64x32
TEXTURE_SIZE2 = ,32;   //Set the width of the texture part width
TEXTURE_SIZE3 = 64;    //Set the height of the texture part height
```

COLOR n = R,G,B,A

Set the badge color .

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

It can be set in the range of n = [1..7].

CONTENTS = { control define … }

It associates the controls to be activated when the button is on.

Automatically the contents hide if a button is the off state. The contents is possible to include all kind of the control.

```
CONTENTS = {
    BUTTON(A) {
        :
    }
    TEXTURE(B) {
        :
    }
    LISTBOX(C) {
        :
    }
}
```

Method of accessing content using C # : [here](#)

GROUP = ID0, ID1, …, IDn

When you associate this control and radio buttons to each other , and one of the group is turned on, the other radio button is automatically turned off.

```
GRUOP = RADIO(A), RADIO(B), RADIO(C);
```

When RADIO (A) is turned on, automatically RADIO (B), RADIO (C) is turned off.

**** STYLE ** = style 0 | style 1 | .. | style n**

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

You set the caption position to the center.

Text anchor	Description
TEXT_CENTER	It aligns the text to the center.
TEXT_LEFT	It aligns the text to the left side.
TEXT_RIGHT	It aligns the text to the right side.

Change the font decoration of the caption .

You can change the font decoration of the caption .	Description
TEXT_NORMAL	No decoration
TEXT_BOLD	Bold type
TEXT_DENT	Dent type
TEXT_SHADOW	Shadow Type

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.

10.13 CHECKBOX

C#: *CWinCtrlCheckbox*

This is a checkbox button control .

It can display seven badges in the same way as buttons .

It is possible to have *contents*. It associates the controls to be activated when the button is on.

```
CHECKBOX(Control name) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.13.1 Example

```
WINDOW(255_000_00001) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,-100;
    TEX_ID = 100_000_00001;
    CAPTION = 000_000_00010;
    SIZE = 512,256;
};

CHECKBOX(TEST) {
    ID = 000_002_00001;
    CAPTION = 000_000_00006;
    STYLE = ANCHOR_CENTER;
    POSITION = 0,-32;
    FONT_KIND= "fn40";
    TEX_ID1 = "bd01";
    TEXTURE_OFFSET1 = 0,10;
    SIZE = 256;
    CONTENTS = {
        ICON(TEST) {
            ID = 000_001_00010;
            POSITION = 256,-16;
            SIZE = 80,80;
            TEX_ID = 100_000_00001,"CRYS";
        }
    }
};
```

```
    };  
}  
};
```



if the button state is on , ICON (TEST) is active .

if the button state is off , ICON (TEST) is not active .

10.13.2 Property

Default value

```
TEX_ID = "BTNO?";  
COLOR = 1,1,1,1;  
CAPTION_COLOR = 1,1,1,1;
```

Part ID rule

It replaces the fifth character of the part ID to 0/1.

- XXXX0: off state
- XXXX1: on state

```
TEX_ID = "BTNO?";
```

If parts ID has been set in this way, you need to prepare parts, such as the following.

- BNT00: off state
- BNT01: on state

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position . By the setting of STYLE, you can change the criteria for display position.

```
POSITION = 32,-64;
```

SIZE = width,height

It changes the display size. If it is omitted , or , you specify 0 ,it gets the size from the texture parts.

```
SIZE = 64,32; //64x32
SIZE = ,32;   //Set the width of the texture part width
SIZE = 64;      //Set the height of the texture part height
```

TEX_ID = Texture ID,Part ID

TEX_ID = Part ID

You set the texture ID and the part ID.

COLOR = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

CAPTION = Caption ID

Set the ID of the string .

Please refer to [here](#) .

CAPTION = “string”

Set the string.

```
CAPTION = "Hello world!";
```

CAPTION_COLOR = R,G,B,A

You can set the caption color .

Set a value in the range of 0~1.

CAPTION_OFFSET = X, Y

It adjust the caption position.

TEX_ID n = Texture ID,Part ID**TEX_ID n = Part ID**

Set the texture ID and the part ID of the badge.

It can be set in the range of n = [1..7].

TEXTURE_OFFSET n = offset X,offset Y

Set the display offset of the badge.

It can be set in the range of n = [1..7].

TEXTURE_SIZE n = width,height

Set the texture ID and the part ID of the badge.

It can be set in the range of n = [1..7].

```
TEXTURE_SIZE1 = 64,32; //64x32
TEXTURE_SIZE2 = ,32;   //Set the width of the texture part width
TEXTURE_SIZE3 = 64;    //Set the height of the texture part height
```

COLOR n = R,G,B,A

Set the badge color .

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

It can be set in the range of n = [1..7].

SE_ID = SE_ID

Set the ID of the sound to sound when pressed . The default is clickSE. The sound will not sound when set to 0 .

CONTENTS = { control define ... }

It associates the controls to be activated when the button is on.

Automatically the contents hide if a button is the off state. The contents is possible to include all kind of the control.

```
CONTENTS = {
    BUTTON(A)  {
        :
    }
    TEXTURE(B) {
        :
    }
    LISTBOX(C) {
        :
    }
}
```

Method of accessing content using C # : [here](#)

**** STYLE ** = style 0 | style 1 | .. | style n**

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

You set the caption position to the center.

Text anchor	Description
TEXT_CENTER	It aligns the text to the center.
TEXT_LEFT	It aligns the text to the left side.
TEXT_RIGHT	It aligns the text to the right side.

Change the font decoration of the caption .

You can change the font decoration of the caption .	Description
TEXT_NORMAL	No decoration
TEXT_BOLD	Bold type
TEXT_DENT	Dent type
TEXT_SHADOW	Shadow Type

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.

10.14 TEXTURE

C#: *CWinCtrlTexture*

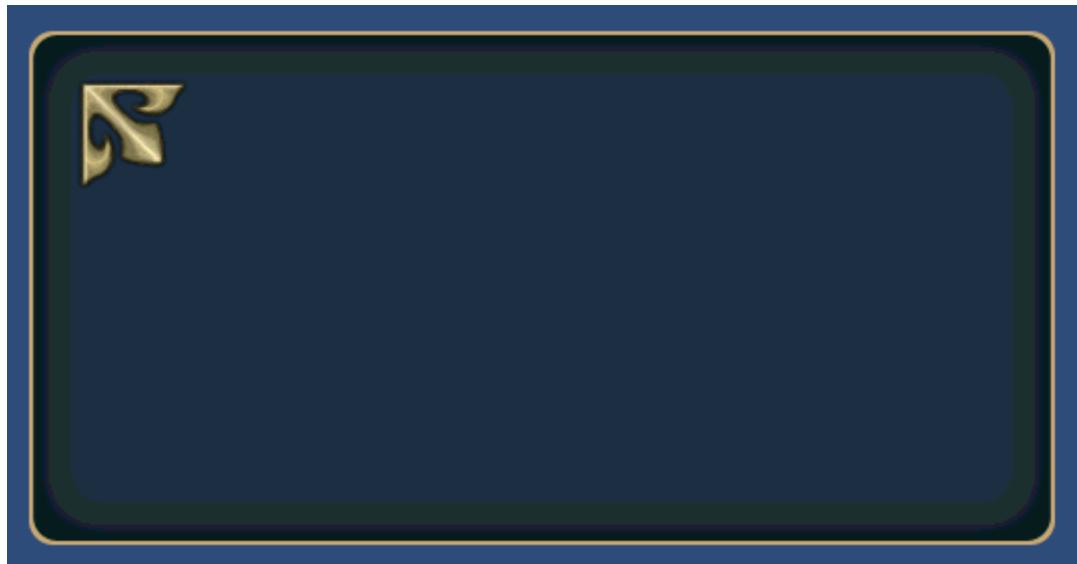
This is a control for displaying a texture part .

It is possible to display eight texture parts.

```
TEXTURE(Control name) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.14.1 Example

```
WINDOW(255_000_00001) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,-100;
    TEX_ID = 100_000_00001;
    TEXTURE_ZOFFSET = 100_010_00000,1;
    SIZE = 512,256;
};
TEXTURE(DecoLeft) {
    ID = 000_000_00040;
    STYLE = NOHIT;
    TEX_ID = 100_010_00000,"DECOL";
    POSITION = 24,-24;
};
```



10.14.2 Property

Default value

```
COLOR = 1,1,1,1;
CAPTION_COLOR = 1,1,1,1;
SE_ID = 0;
```

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position . By the setting of STYLE, you can change the criteria for display position.

```
POSITION = 32,-64;
```

TEX_ID = Texture ID,Part ID

TEX_ID = Part ID

You set the texture ID and the part ID.

TEX_ID n = Texture ID,Part ID

TEX_ID n = Part ID

You set the texture ID and the part ID.

It can be specified in the range of n = [0..7].

If n = 0, it operates the same texture as the TEX_ID.

TEXTURE_OFFSET n = offset X,offset Y

It defines the display offset of the texture .

It can be specified in the range of n = [0..7].

SIZE = width,height

It changes the display size. If it is omitted , or , you specify 0 ,it gets the size from the texture parts.

```
SIZE = 64,32; //64x32
SIZE = ,32;   //Set the width of the texture part width
SIZE = 64;     //Set the height of the texture part height
```

TEXTURE_SIZE n = width,height

You set the texture ID and the part ID.

It can be specified in the range of n = [0..7].

when n = 0 is the same meaning as ** SIZE **.

```
TEXTURE_SIZE1 = 64,32; //64x32
TEXTURE_SIZE2 = ,32;   //Set the width of the texture part width
TEXTURE_SIZE3 = 64;     //Set the height of the texture part height
```

COLOR = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

COLOR n = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

It can be specified in the range of n = [0..7].

If n = 0, it operates the same texture as COLOR .

SE_ID = SE_ID

To set the ID of the sound played when pressed . 0 is set as the default. This default value means not to play a sound.

STYLE = Flag0|Flag1|..|Flagn

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.

10.15 LINE

C#: *CWinCtrlLine*

This is the control to draw a line segment.

```
LINE(Control name) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.15.1 Example

```
WINDOW(255_000_00001) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_NOFRAME|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,-100;
    TEX_ID = 100_010_00000;
    SIZE = 512,256;
};

LINE(TEST) {
    ID = 000_000_00040;
    TEX_ID = "LINE0";
    POSITION = 24,-24;
    POSITION2 = 256,-128;
};
```



10.15.2 Property

Default value

```
TEX_ID = "LINE";
COLOR = 1,1,1,1;
CAPTION_COLOR = 1,1,1,1;
```

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the starting position .

```
POSITION = 32,-64;
```

POSITION2 = X, Y

Determine the end position.

```
POSITION = 32,-64;
```

TEX_ID = Texture ID,Part ID

TEX_ID = Part ID

You set the texture ID and the part ID.

SIZE = Segment length,Segment thickness

Automatically the window system is substituted the length of the line segment into X.

Set the thickness of the line segment in the Y.

```
SIZE = ,32; //line width = 32
```

COLOR = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

**** STYLE ** = style 0 | style 1 | .. | style n**

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.

10.16 RENDER

C#: *CWinCtrlRender*

It is a control for the Render Texture.

When the control is generated, generates the camera together.

Objects rendered on the camera is drawn.

Note: if you have loaded resources, you need to delete them when you close window.

If you forget to delete them, your application memory will leak.

```
RENDER(Control name) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.16.1 Example

```
WINDOW(255_000_00001) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,-100;
    SIZE = 512,256;
};

RENDER(AVATAR) {
    ID = 000_000_00200;
    STYLE = ANCHOR_CENTER;
    POSITION = 0,0;
    SIZE = RELATIVE_SIZE(-64),RELATIVE_SIZE(-64);
    CONTENTS_SIZE = 512,512;
};
```

10.16.2 Property

Default value

```
COLOR = 1,1,1,1;
SE_ID = 0;
```

**ID = Control ID**

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position . By the setting of STYLE, you can change the criteria for display position.

```
POSITION = 32, -64;
```

SIZE = width,height

Set the display size.

```
SIZE = 640, 480;           //display size
```

Note: Properties that must always be set.

CONTENTS_SIZE = width,height

Set the size of the Render Texture.

```
CONTENTS_SIZE = 1024, 1024;      //render texture size
```

Note: It is necessary to set a power value of two and set square. Also, by setting the 2048 value less equal than the maximum size , it increases compatibility.

COLOR = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

COLOR1 = R,G,B,A

Set the background color.

Set R, G, B, A between 0-1

SE_ID = SE_ID

To set the ID of the sound played when pressed . 0 is set as the default. This default value means not to play a sound.

TEX_ID = Render texture ID

It is used as a name of the Render Texture. If you set this value to TEXTURE_ZOFFSET of WINDOW property , it is possible to control the rendering order.

STYLE = Flag0|Flag1|..|Flagn

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.

10.17 ICON

C#: *CWinCtrlIcon*

This is the control to display the icon.

It is possible to display eight texture parts.

This behavior is almost the same as *TEXTURE*. But if you touch it, it becomes darker.

The different points are that its color becomes dark when you touch, and it can have a caption.

```
ICON(Control name) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.17.1 Example

```
WINDOW(001_002_00003) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,-100;
    SIZE = 512,256;
};

ICON(TEST) {
    ID = 000_001_00010;
    POSITION = 256,-24;
    SIZE = 80,80;
    TEX_ID = 100_000_00001,"BNADO";
    TEX_ID1 = 100_000_00001,"bd01";
    TEX_ID3 = 100_000_00001,"CRYs";
};
```



10.17.2 Property

Default value

```
COLOR = 1,1,1,1;
CAPTION_COLOR = 1,1,1,1;
SE_ID = 0;
```

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position . By the setting of STYLE, you can change the criteria for display position.

```
POSITION = 32, -64;
```

TEX_ID = Texture ID,Part ID**TEX_ID = Part ID**

You set the texture ID and the part ID.

TEX_ID n = Texture ID,Part ID**TEX_ID n = Part ID**

You set the texture ID and the part ID.

It can be specified in the range of n = [0..7].

If n = 0, it operates the same texture as the TEX_ID.

TEXTURE_OFFSET n = offset X,offset Y

It changes the display size. If it is omitted , or , you specify 0 ,it gets the size from the texture parts.

It can be specified in the range of n = [0..7].

if you did not set or if n = 0,it is the same as setting the TEX_ID.

SIZE = width,height

It changes the display size. If it is omitted , or , you specify 0 ,it gets the size from the texture parts.

```
SIZE = 64,32; //64x32
SIZE = ,32;   //Set the width of the texture part width
SIZE = 64;     //Set the height of the texture part height
```

TEXTURE_SIZE n = width,height

You set the texture ID and the part ID.

It can be specified in the range of n = [0..7].

If n = 0, it operates the same texture as SIZE .

```
TEXTURE_SIZE1 = 64,32; //64x32
TEXTURE_SIZE2 = ,32; //Set the width of the texture part width
TEXTURE_SIZE3 = 64; //Set the height of the texture part height
```

COLOR = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

SE_ID = SE_ID

To set the ID of the sound played when pressed . 0 is set as the default. This default value means not to play a sound.

COLOR n = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

It can be specified in the range of n = [0..7].

If n = 0, it operates the same texture as **COLOR** .

CAPTION = Caption ID

Set the ID of the string .

Please refer to [here](#) .

CAPTION = “string”

Set the string.

```
CAPTION = "Hello world!";
```

CAPTION_COLOR = R,G,B,A

You can set the caption color .

Set a value in the range of 0~1.

CAPTION_OFFSET = X, Y

It adjust the caption position.

STYLE = Flag0|Flag1|..|Flagn

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

You set the caption position to the center.

Text anchor	Description
TEXT_CENTER	It aligns the text to the center.
TEXT_LEFT	It aligns the text to the left side.
TEXT_RIGHT	It aligns the text to the right side.

Change the font decoration of the caption .

You can change the font decoration of the caption .	Description
TEXT_NORMAL	No decoration
TEXT_BOLD	Bold type
TEXT_DENT	Dent type
TEXT_SHADOW	Shadow Type

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.

10.18 RECASTICON

C#: *CWinCtrlRecastIcon*

It is a control to display a recast icon.

It is possible to display eight texture parts.

This is almost the same behavior as the ICON.

If you make a set of recast value, the specified texture parts do animation which it becomes gradually extending from the bottom. If you set 0 to display as TEXTURE. If you set 1 or more, it hides.

10.18.1 Example

```
WINDOW(255_000_00001) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_NOFRAME|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,-100;
    SIZE = 512,256;
};

RECASTICON(Test) {
    ID = 000_000_00010;
    POSITION = 5,5;
    STYLE = ANCHOR_LEFTBOTTOM|INPUT_NOBLOCK;
    TEX_ID0 = 100_200_00000,"btrc";
    TEX_ID1 = 100_200_00000,"SKLWT";
    TEX_ID2 = 100_200_00000,"aF01";
    SIZE = 70,70;
    TEXTURE_OFFSET1 = 5,-5;
    TEXTURE_OFFSET2 = 6,-6;
    SE_ID = 0;
    PRIORITY = 1;
};
```



10.18.2 Property

Default value

```
COLOR = 1,1,1,1;
CAPTION_COLOR = 1,1,1,1;
SE_ID = 0;
```

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position . By the setting of STYLE, you can change the criteria for display position.

```
POSITION = 32,-64;
```

TEX_ID = Texture ID,Part ID

TEX_ID = Part ID

You set the texture ID and the part ID.

TEX_ID n = Texture ID,Part ID

TEX_ID n = Part ID

You set the texture ID and the part ID.

It can be specified in the range of n = [0..7].

If n = 0, it operates the same texture as the TEX_ID.

TEXTURE_OFFSET n = offset X,offset Y

It changes the display size. If it is omitted , or , you specify 0 ,it gets the size from the texture parts.

It can be specified in the range of n = [0..7].

if you did not set or if n = 0,it is the same as setting the TEX_ID.

SIZE = width,height

It changes the display size. If it is omitted , or , you specify 0 ,it gets the size from the texture parts.

```
SIZE = 64,32; //64x32
SIZE = ,32;   //Set the width of the texture part width
SIZE = 64;     //Set the height of the texture part height
```

TEXTURE_SIZE n = width,height

You set the texture ID and the part ID.

It can be specified in the range of n = [0..7].

If n = 0, it operates the same texture as **SIZE** .

```
SIZE = 64,32; //64x32
SIZE = ,32;   //Set the width of the texture part width
SIZE = 64;     //Set the height of the texture part height
```

COLOR = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

COLOR n = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

It can be specified in the range of n = [0..7].

If n = 0, it operates the same texture as **COLOR** .

FONT_KIND = Font kind

Set the font kind

```
FONT_KIND = "fnt32";
```

CAPTION = Caption ID

Set the ID of the string .

Please refer to [here](#) .

```
CAPTION = 010_000_00100;
```

CAPTION = "string"

Set the string.

```
CAPTION = "Hello world!";
```

CAPTION_COLOR = R,G,B,A

You can set the caption color .

Set a value in the range of 0~1.

CAPTION_OFFSET = X, Y

It adjust the caption position.

SE_ID = SE_ID

To set the ID of the sound played when pressed . 0 is set as the default. This default value means not to play a sound.

STYLE = Flag0|Flag1|..|Flagn

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

You set the caption position to the center.

Text anchor	Description
TEXT_CENTER	It aligns the text to the center.
TEXT_LEFT	It aligns the text to the left side.
TEXT_RIGHT	It aligns the text to the right side.

Change the font decoration of the caption .

You can change the font decoration of the caption .	Description
TEXT_NORMAL	No decoration
TEXT_BOLD	Bold type
TEXT_DENT	Dent type
TEXT_SHADOW	Shadow Type

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.

10.19 RENDERICON

C#: *CWinCtrlRenderIcon*

It is a control for the icon with using Render Texture.

It automatically cut out rectangle from the Render Texture area which you specify.

Multiple rendering result may be included within single Render Texture. After you render to the texture,it can render many icons at low cost.

For example ,it is effective to use that when you want to display the dress possible avatar icon on the listbox.

RENDERICON(Control name)

Property 1;

Property 2;

:

```
:
Property n
};
```

10.19.1 Example

```
RENDERICON(Avatar) {
    ID = 000_000_00330;
    POSITION = -315,-22;
    TEX_ID = 255_000_00010;           //render texture id
    CONTENTS_SIZE = 1024,1024;      //render texture size
    SIZE = 64,64;                  //display size
};
```

In this example, you can be sharing the 256(= 1024/64 x 1024/64) icon on a single texture. And it can be rendered by the single mesh. However, doing the sharing of more than 256 , it fails to allocate the area, and can't render.

10.19.2 Property

Default value

```
COLOR = 1,1,1,1;
SE_ID = 0;
```

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position . By the setting of STYLE, you can change the criteria for display position.

```
POSITION = 32,-64;
```

TEX_ID = Texture ID

Set the ID of the Render Texture.If you set the same ID as the other RENDERICON ,they share the same Render Texture.In this case, it is allocated by dividing the Render Texture to the icon size.

By setting this value to *TEXTURE_ZOFFSET = Texture ID,Z offset* , it is possible to control the rendering order .

A Render Texture size to be allocated is CONTENTS_SIZE.

Note: Properties that must always be set.

Note: Share the Render Texture only in the same window . Render Textures are not shared across a window.

CONTENTS_SIZE = Render Texture size,Render Texture size

Set the size of the Render Texture . It is necessary to set a common value between icons to share.

When you set a different value , there is no guarantee of behaviour.

Note: Properties that must always be set.

Set the same value between Render Texture that you want to share.

You can increase the compatibility if you note the following. the Render Texture size sets a power of two, sets square, and sets the less than or equal to 2048.

TEXTURE_OFFSET = offset X,offset Y

Set the display offset.

SIZE = width,height

Set the display size.

And it is used to determine the allocation size from the Render Texture.

Error occurs when you set a different value between the share to that icon.

```
SIZE = 64,32; //display size
```

Note: Properties that must always be set.

Set the same value between Render Texture that you want to share.

COLOR = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

COLOR1 = R,G,B,A

Set the background color.

Set R, G, B, A between 0-1

SE_ID = SE_ID

To set the ID of the sound played when pressed . 0 is set as the default. This default value means not to play a sound.

STYLE = Flag0|Flag1|..|Flagn

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.

10.20 METER

C#: *CWinCtrlMeter*

This is the control to display a meter.

You can use this as a progress bar.

It is possible to display eight texture parts.

You can stretch any texture parts.

If the length is set to 1, it display same look as *TEXTURE* .

```
METER(Control name) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.20.1 Example

```
WINDOW(255_000_00001) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,-100;
    TEX_ID = 100_000_00001;
    CAPTION = 000_000_00010;
    SIZE = 512,256;
};

METER(Progress) {
    ID = 000_001_00000;
    STYLE = WIN_CTRL_STYLE_ANCHOR_CENTER;
    SIZE = RELATIVE_SIZE(-128);
    POSITION = 0,0;
    TEX_ID = 0,"MTR";
    COLOR = 1,1,1,1;
    TEX_ID1 = 0,"MTRB";
    COLOR1 = 1,1,1,1;
};
```

10.20.2 Property

Default value



```
TEX_ID0 = "MTR";
TEX_ID1 = "MTRB";
COLOR = 1,1,1,1;
CAPTION_COLOR = 1,1,1,1;
```

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position . By the setting of STYLE, you can change the criteria for display position.

```
POSITION = 32,-64;
```

TEX_ID = Texture ID,Part ID

TEX_ID = Part ID

You set the texture ID and the part ID.

TEX_ID n = Texture ID,Part ID

TEX_ID n = Part ID

You set the texture ID and the part ID.

It can be specified in the range of n = [0..7].

If n = 0, it operates the same texture as the TEX_ID.

TEXTURE_OFFSET n = offset X,offset Y

It changes the display size. If it is omitted , or , you specify 0 ,it gets the size from the texture parts.

It can be specified in the range of n = [0..7].

If you did not set or if n = 0,it is the same as setting the TEX_ID.

SIZE = width,height

It changes the display size. If it is omitted , or , you specify 0 ,it gets the size from the texture parts.

```
SIZE = 64,32; //64x32
SIZE = ,32;   //Set the width of the texture part width
SIZE = 64;     //Set the height of the texture part height
```

TEXTURE_SIZE n = width,height

You set the texture ID and the part ID.

It can be specified in the range of n = [0..7].

If n = 0, it operates the same texture as **SIZE** .

```
TEXTURE_SIZE1 = 64,32; //64x32
TEXTURE_SIZE2 = ,32;   //Set the width of the texture part width
TEXTURE_SIZE3 = 64;     //Set the height of the texture part height
```

COLOR = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

COLOR n = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

It can be specified in the range of n = [0..7].

If n = 0, it operates the same texture as **COLOR** .

STYLE = Flag0|Flag1|..|Flagn

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.

10.21 SCROLLBAR

C#: *CWinCtrlScrollbar*

This is a control to display the scroll bar.

By assigning to the scrollable control, this control is enabled.

By associating SCROLLBAR control the GROUP property, it is possible to display a scroll bar.

It is also possible to assign some scrollbars to a scrollable control.

It is listed scrollable controls below.

- *LISTBOX*
- *LISTBOXEX*
- *EDITBOX*
- *TEXTBOX*
- *CONTAINER*
- *LOG*

```
SCROLLBAR(Control name) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.21.1 Example

```
$w = 300;
LISTBOX(List) {
    ID = 001_100_00000;
    POSITION = 0,160;
    STYLE = ANCHOR_BOTTOM;
    SIZE = $w,160 - 16;
    CONTENTS_SIZE = $w,48;
    CONTENTS = {
        CHECKBOX(IP) {
            ID = 001_000_00020;
            CAPTION = 001_000_00030;
            STYLE = ANCHOR_LEFTTOP;
            POSITION = 0,0;
            SIZE = $w,48;
        };
    }
    GROUP = SCROLLBAR(List);
};

SCROLLBAR(List) {
    ID = 001_100_00010;
    DEF_SCROLLBAR;
    STYLE = ANCHOR_RIGHTTOP;
    POSITION = 0,-16;
    SIZE = 0,- 16 * 2;
};
```



10.21.2 Position and size

Unlike normal control, handling of location and size of this control is special.

The position and size are relative to the scrollable controls that are assigned.

Position and size is a relative value from the scrollable are related control.

Valid anchors to see [here](#).

```
#include "wr.h"

WINDOW(255_000_00001) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,0;
    SIZE = 256,256;
    TEX_ID = 100_000_00001;
};

CONTAINER(Map) {
    ID = 000_000_000100;
    POSITION = 0,0;
    SIZE = RELATIVE_SIZE(0),RELATIVE_SIZE(0);
    CONTENTS_SIZE = 512,512;
    COLOR = COLOR32(255,255,255,255);
    CONTENTS = {
        BUTTON(A) {
            STYLE = BASE_LEFT;
            PRIORITY = -1;
            POSITION = 0,-256;
            TEX_ID = "TQML0";
        };
        BUTTON(B) {
            STYLE = BASE_TOP;
            PRIORITY = -1;
            POSITION = 256,0;
            TEX_ID = "TQML0";
        };
    };
};
```

```

BUTTON(C) {
    STYLE = BASE_BOTTOM;
    PRIORITY = -1;
    POSITION = 256,-512;
    TEX_ID = "TQML0";
};

BUTTON(D) {
    STYLE = BASE_RIGHT;
    PRIORITY = -1;
    POSITION = 512,-256;
    TEX_ID = "TQML0";
};

GROUP = SCROLLBAR(LEFT),SCROLLBAR(RIGHT),SCROLLBAR(TOP),SCROLLBAR(BOTTOM);
};

SCROLLBAR(LEFT) {
    STYLE = ANCHOR_LEFTTOP;
    TEX_ID = "SCBBH";
    POSITION = 0,0;
    SIZE = 0,0;
};

SCROLLBAR(RIGHT) {
    STYLE = ANCHOR_RIGHTTOP;
    TEX_ID = "SCBBH";
    POSITION = 0,0;
    SIZE = 0,0;
};

SCROLLBAR(TOP) {
    STYLE = ANCHOR_LEFTTOP|LISTBOX_ITEM_STACK_H;
    TEX_ID = "SCBBV";
    POSITION = 0,0;
    SIZE = 0,0;
};

SCROLLBAR(BOTTOM) {
    STYLE = ANCHOR_LEFTBOTTOM|LISTBOX_ITEM_STACK_H;
    TEX_ID = "SCBBV";
    POSITION = 0,0;
    SIZE = 0,0;
};

```

10.21.3 Direction of the scroll bar

You can use not only the vertical scroll bar but also the horizontal scroll bar.

Please set the following values to the style.

- Vertical direction: LISTBOX_ITEM_STACK_V
- Horizontal direction: LISTBOX_ITEM_STACK_H

10.21.4 Display control of the scroll bar.

By setting the style, you can choose the scrollbar display type whether display or hide.

You can choose from the following three types.

SCROLLBAR_DISPLAY_NORMAL(Default)

Only when the area is scrollable and you touch the scrollable area,it is displayed.

SCROLLBAR_DISPLAY_SCROLLABLE

It display only when the area is scrollable.

SCROLLBAR_DISPLAY_ALWAYS

Always displays.

10.21.5 Property**Default value**

```
TEX_ID = "SCBRH";
STYLE = LISTBOX_ITEM_STACK_V;
COLOR = 1,1,1,1;
```

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position. It becomes relative position from a scrollable control.

```
POSITION = 0,-16;
```

TEX_ID = Texture ID,Part ID**TEX_ID = Part ID**

You set the texture ID and the part ID.

SIZE = width,height

Set the display size of the scroll bar. It is the relative size of the scrollable control.

```
SIZE = 0,0; //display size
```

COLOR = R,G,B,A

It is possible to set the color values . By changing color , it affects all of the controls that are included .

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

STYLE = Flag0|Flag1|..|Flagn

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

The actual placement, please refer to the figure below.



Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
LISTBOX_ITEM_STACK_V	Set to the direction of the scroll bar in the vertical direction.
LISTBOX_ITEM_STACK_H	Set to the direction of the scroll bar in the horizontal direction.
SCROLLBAR_DISPLAY_NORMAL	Only when the area is scrollable and you touch the scrollable area,it is displayed.
SCROLLBAR_DISPLAY_SCROLLABLE	It display only when the area is scrollable.
SCROLLBAR_DISPLAY_ALWAYS	Always displays.

10.22 LISTBOX

C#: *CWinCtrlListbox*

This is the control for the list box.

It has contents of the same number as the number of rows.

It can have a lot of contents as long as the remaining memory.

Not only the list box that is lined with content in the vertical direction , it is possible to arrange also in the horizontal direction .

```
LISTBOX(Control name) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.22.1 Example

```
WINDOW(255_000_00001) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_NOFRAME|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,-100;
    SIZE = 512,256;
};

$w = 300;
LISTBOX(List) {
    ID = 001_100_00000;
    POSITION = 0,160;
    STYLE = ANCHOR_BOTTOM;
    SIZE = $w,160 - 16;
    CONTENTS_SIZE = $w,48;
    CONTENTS = {
        CHECKBOX(IP) {
            ID = 001_000_00020;
            CAPTION = 001_000_00030;
            STYLE = ANCHOR_LEFTTOP;
            POSITION = 0,0;
            SIZE = $w,48;
        };
    }
    GROUP = SCROLLBAR(List);
};
SCROLLBAR(List) {
    ID = 001_100_00010;
    DEF_SCROLLBAR;
    STYLE = ANCHOR_RIGHTTOP;
    POSITION = 0,-16;
    SIZE = 0,-16 * 2;
};
```



10.22.2 Property

Default value

```
STYLE = LISTBOX_ITEM_STACK_V; //Arranged in the vertical direction
COLOR = 1,1,1,1;
CAPTION_COLOR = 1,1,1,1;
SE_ID = scrollSE;
```

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position.

```
POSITION = 32,-64;
```

SIZE = width,height

Set the display area of the list box.

Control protruding from the display area is clipped .

```
SIZE = 400,400; //display size
CONTENTS_SIZE = 800,80; //one contents size
```

CONTENTS_SIZE = width, height

Determine the listbox size of the line. A listbox which arranged in the vertical direction, scrolls in the vertical direction.

However ,if the width of the content size is larger than the display area, the listbox scrolls to the left or right.

```
// only vertical direction to scroll
SIZE = 400,400;           //display size
CONTENTS_SIZE = 400,40;    //virtual screen size

// Scroll to the left and right direction
SIZE = 400,400;           //display size
CONTENTS_SIZE = 800,40;    //virtual screen size
```

CONTENTS = { control define ... }

Enumerate the controls for defining a line in the LISTBOX.

A position of the controls place with a relative from the position of the *LISTBOX*.

It is possible to include any type of control .

```
// It is also possible to put the list box in the list box
CONTENTS = {
    BUTTON(A) {
        :
    };
    CHECKBOX(B) {
        :
    };
    LISTBOX(C) {
        :
    };
}
```

How to access to a particular control of the content list from C#, please refer to [here](#).

LINE_SPACE = line spacing pixel value

It defines the gaps between the content.

```
LINE_SPACE = 8;           //Put an 8-dot space.
```

GROUP = SCROLLBAR Control ID,...

If you set scroll controls to the GROUP property,it display scroll bars in conjunction with the listbox.

It should be careful not to set to CONTENTS.

It is also possible to assign multiple scrollbars.

```
GROUP = SCROLLBAR(Horizon),SCROLLBAR(Vertical);
```

COLOR = R,G,B,A

It is possible to set the color values . By changing color , it affects all of the controls that are included .

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

SE_ID = SE_ID

When the head of the content has changed during the scroll , set the ID of the sound effect. The default is “scrollSE”. if it is 0 , sound should not be.

STYLE = Flag0|Flag1|..|Flagn

Configurable unique styles in the list box are like the following .

Listbox control flag	Description
LISTBOX_ITEM_STACK_V	List box arranged in the vertical direction.
LISTBOX_ITEM_STACK_H	List box arranged in the horizontal direction.
WIN_CTRL_STYLE_NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.
LISTBOX_LOCK	Suppress that scroll the listbox.

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.
LISTBOX_ITEM_STACK_V	List box arranged in the vertical direction.
LISTBOX_ITEM_STACK_H	List box arranged in the horizontal direction.
WIN_CTRL_STYLE_NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.
LISTBOX_LOCK	Suppress that scroll the listbox.

10.23 LISTBOXEX

C#: *CWinCtrlListboxEx*

This is the control for the list box.

It has contents of the same number as the number of rows.

It can have a lot of contents as long as the remaining memory.

Not only the list box that is lined with content in the vertical direction , it is possible to arrange also in the horizontal direction .

Differences between the *LISTBOX* is to measure the content size automatically.

If the content is arranged in a vertical direction, it automatically calculate only height.

If the content is arranged in a horizontal direction, it automatically calculate only width.

```
LISTBOXEX(Control name) {
```

```
    Property 1;
```

```
    Property 2;
```

```
:
```

```
:
```

```
    Property n
```

```
};
```

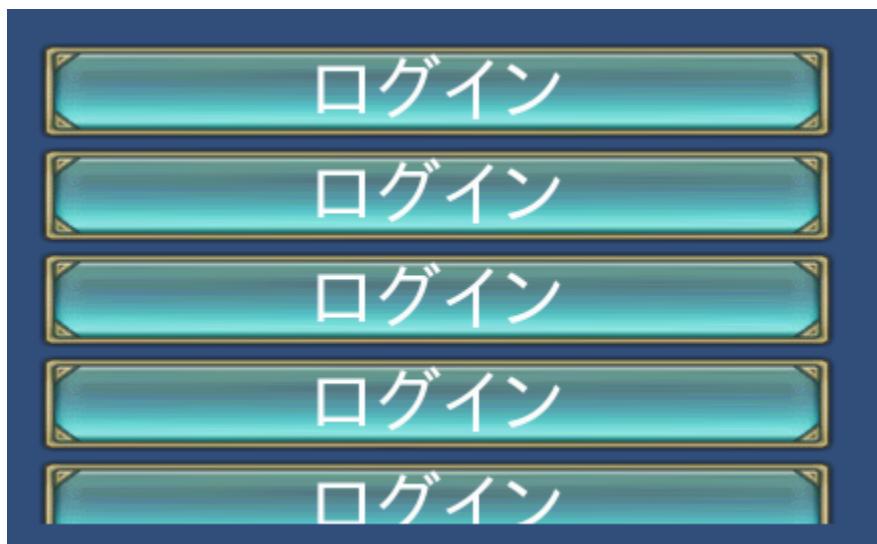
10.23.1 Example

```

WINDOW(255_000_00001) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_NOFRAME|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,-100;
    SIZE = 512,256;
};

$w = 400;
LISTBOXEX(List) {
    ID = 001_100_00000;
    POSITION = 0,0;
    STYLE = ANCHOR_BOTTOM;
    SIZE = $w,RELATIVE_SIZE(0) - 16;
    CONTENTS = {
        CHECKBOX(IP) {
            ID = 001_000_00020;
            CAPTION = 001_000_00030;
            STYLE = ANCHOR_LEFTTOP;
            POSITION = 0,0;
            SIZE = $w,48;
        };
    };
    LINE_SPACE = 4;
    GROUP = SCROLLBAR(List);
};
SCROLLBAR(List) {
    ID = 001_100_00010;
    STYLE = ANCHOR_LEFTTOP;
    POSITION = $w + 4,-16;
    SIZE = 0,160 - 16 - 16 * 2;
};

```



10.23.2 Property

Default value

```

STYLE = LISTBOX_ITEM_STACK_V; //Arranged in the vertical direction
COLOR = 1,1,1,1;

```

```
CAPTION_COLOR = 1,1,1,1;
SE_ID = scrollSE;
```

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position.

```
POSITION = 32,-64;
```

SIZE = width,height

Set the display area of the list box.

Control protruding from the display area is clipped .

```
SIZE = 400,400;           //display size
CONTENTS_SIZE = 800,80;    //one contents size
```

CONTENTS_SIZE = width, height

Set the size of the line.

If the content is arranged in a vertical direction , the height of the content is automatically calculated. Therefore this property affects only the width of the content .

On the other hand,if the content is arranged in a horizontal direction , the width of the content is automatically calculated. Therefore this property affects only the height of the content .

Otherwise, it is the same as the [LISTBOX](#).

```
// only vertical direction to scroll
SIZE = 400,400;           //display size
CONTENTS_SIZE = 400,40;    //virtual screen size

// Scroll to the left and right direction
SIZE = 400,400;           //display size
CONTENTS_SIZE = 800,40;    //virtual screen size
```

CONTENTS = { Control define; ... }

Set controls to define a line of LISTBOX.

A position of the controls place with a relative from the position of the [LISTBOX](#).

It is possible to include any type of control .

```
// It is also possible to put the list box in the list box
CONTENTS = {
    BUTTON(A) {
        :
    }
}
```

```

};

CHECKBOX (B) {
:
};

LISTBOX (C) {
:
};

```

How to access to a particular control of the content list from C#, please refer to [here](#).

LINE_SPACE = line spacing pixel value

It defines the gaps between the content.

```
LINE_SPACE = 8;           //Put an 8-dot space.
```

GROUP = SCROLLBAR Control ID,...

If you set scroll controls to the GROUP property,it display scroll bars in conjunction with the listbox.

It should be careful not to set to CONTENTS.

It is also possible to assign multiple scrollbars.

```
GROUP = SCROLLBAR(Horizon), SCROLLBAR(Vertical);
```

COLOR = R,G,B,A

It is possible to set the color values . By changing color , it affects all of the controls that are included .

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

SE_ID = SE_ID

When the head of the content has changed during the scroll , set the ID of the sound effect. The default is “scrollSE”. if it is 0 , sound should not be.

STYLE = Flag0|Flag1|..|Flagn

Configurable unique styles in the list box are like the following .

Listbox control flag	Description
LISTBOX_ITEM_STACK_V	List box arranged in the vertical direction.
LISTBOX_ITEM_STACK_H	List box arranged in the horizontal direction.
WIN_CTRL_STYLE_NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.
LISTBOX_LOCK	Suppress that scroll the listbox.

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.
LISTBOX_ITEM_STACK_V	List box arranged in the vertical direction.
LISTBOX_ITEM_STACK_H	List box arranged in the vertical direction.
WIN_CTRL_STYLE_NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.
LISTBOX_LOCK	Suppress that scroll the listbox.

10.24 CONTAINER

C#: *CWinCtrlContainer*

It is a control for grouping multiple controls .

Size of the container is determined by the SIZE. The size of the virtual screen is determined by CONTENTS_SIZE.

When virtual screen size is larger than the display size , it is possible to scroll the container.

```
CONTAINER(Control name) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.24.1 Example

```
WINDOW(255_000_00001) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,-100;
    SIZE = 512,256;
    TEX_ID = 100_010_00000;
};

$w = 400;
CONTAINER(Map) {
    ID = 000_000_000100;
    POSITION = 0,0;
    SIZE = RELATIVE_SIZE(0),RELATIVE_SIZE(0);
    CONTENTS_SIZE = 640,1136;
    COLOR = COLOR32(255,255,255,255);
    CONTENTS = {
        TEXTURE(Map) {
            ID = 000_000_000110;
            SIZE = 640,1136;
            PRIORITY = -1;
            POSITION = 0,0;
            TEX_ID = 0,"MAP2";
        };
    };
};
```



10.24.2 Property

Default value

```
COLOR = 1,1,1,1;
CAPTION_COLOR = 1,1,1,1;
```

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position.

```
POSITION = 32,-64;
```

SIZE = width,height

It defines the display area of the container.

Control protruding from the display area is clipped .

```
SIZE = 400,400;          //display size
CONTENTS_SIZE = 800,800;  //virtual screen size
```

CONTENTS_SIZE = width, height

It defines the size of the virtual screen .

When virtual screen size is larger than the display size , it is possible to scroll the container.

```
SIZE = 400,400;           //display size
CONTENTS_SIZE = 800,800;   //virtual screen size
```

CONTENTS = { control define ... }

It lists the control that contained in the CONTAINER.

Coordinates of the controls in the contents is placed in a relative coordinate from the CONTAINER.

It is possible to include any type of control .

```
CONTENTS = {
    BUTTON(A)  {
        :
    };
    CHECKBOX(B) {
        :
    };
    LISTBOX(C) {
        :
    };
};
```

Method of accessing content using C # : [here](#)

COLOR = R,G,B,A

It is possible to set the color values . By changing color , it affects all of the controls that are included .

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

GROUP = SCROLLBAR Control ID,...

If you set scroll controls to the GROUP property,it display scroll bars in conjunction with the container.

It should be careful not to set to CONTENTS.

It is also possible to assign multiple scrollbars.

```
GROUP = SCROLLBAR(Horizon),SCROLLBAR(Vertical);
```

STYLE = Flag0|Flag1|..|Flagn

Container-specific styles are as follows.

Container control flag	Description
WIN_CTRL_STYLE_NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.
WIN_CTRL_STYLE_NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.

10.25 FRAME

C#: *CWinCtrlFrame*

This is the control to display the frame.

Unlike *BAR* , it can not have a caption.

It is possible to receive the onClick event .

FRAME(Control name)

 Property 1;

 Property 2;

 :

 :

 Property n

}

10.25.1 Example

```
WINDOW(255_000_00001) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_NOFRAME|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,-100;
    SIZE = 512,256;
};

FRAME(Test) {
    ID = 000_000_00010;
    POSITION = 0,0;
    SIZE = RELATIVE_SIZE(0),RELATIVE_SIZE(0);
};
```

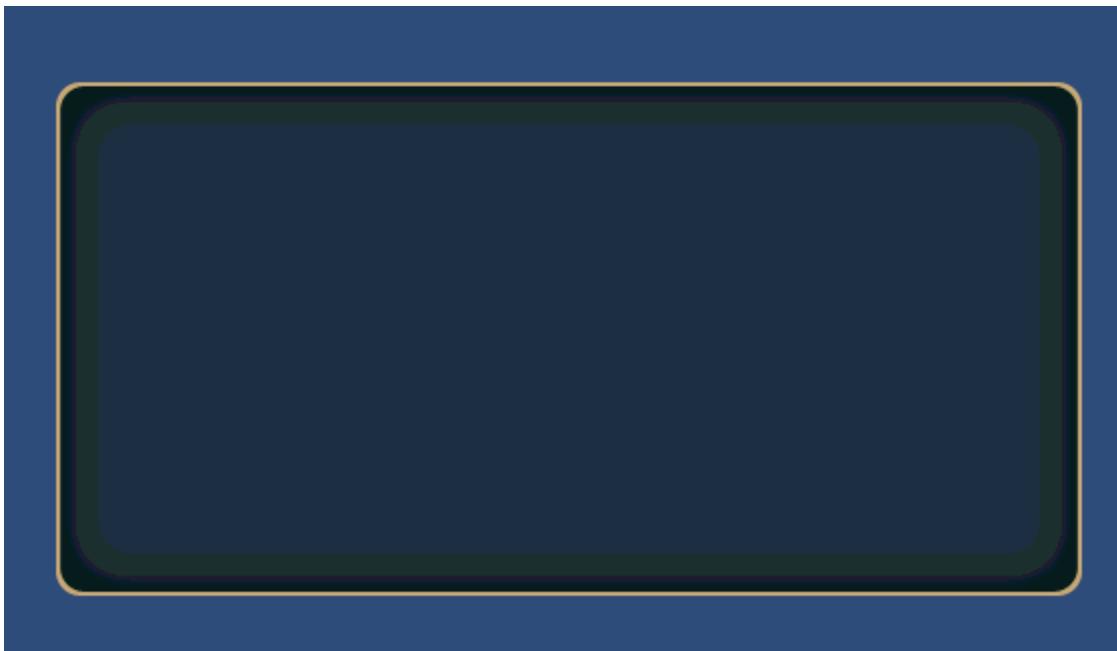
10.25.2 Property

Default value

```
TEX_ID = "FRAME";
COLOR = 1,1,1,1;
CAPTION_COLOR = 1,1,1,1;
SE_ID = 0;
```

ID = Control ID

Define the control ID.



```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position . By the setting of STYLE, you can change the criteria for display position.

```
POSITION = 32,-64;
```

TEX_ID = Texture ID,Part ID

TEX_ID = Part ID

You set the texture ID and the part ID.

SIZE = width,height

It changes the display size. If it is omitted , or , you specify 0 ,it gets the size from the texture parts.

```
SIZE = 64,32; //64x32
SIZE = ,32;   //Set the width of the texture part width
SIZE = 64;     //Set the height of the texture part height
```

COLOR = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

SE_ID = SE_ID

To set the ID of the sound played when pressed . 0 is set as the default. This default value means not to play a sound.

STYLE = Flag0|Flag1|..|Flagn

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.

10.26 LABEL

C#: *CWinCtrlLabel*

This is the control to display a label.

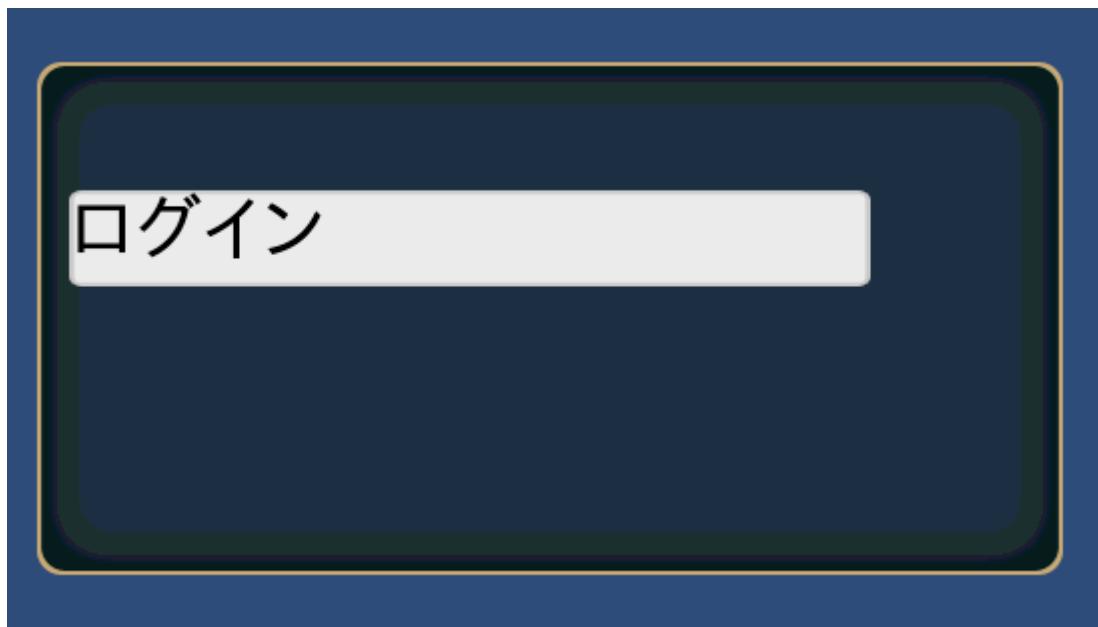
```
LABEL(Control name) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.26.1 Example

```
WINDOW(255_000_00001) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,-100;
    SIZE = 512,256;
};

$w = 400;
LABEL(TEST) {
    ID = 001_000_00020;
    CAPTION = 001_000_00030;
    CAPTION_COLOR = 0,0,0,1;
    STYLE = ANCHOR_LEFTTOP;
    POSITION = 16,-64;
```

```
SIZE = $w, 48;
};
```



10.26.2 Property

Default value

```
TEX_ID = "LABEL";
STYLE = TEXT_CENTER;
COLOR = 1,1,1,1;
CAPTION_COLOR = 1,1,1,1;
```

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position . By the setting of STYLE, you can change the criteria for display position.

```
POSITION = 32,-64;
```

TEX_ID = Texture ID,Part ID

TEX_ID = Part ID

You set the texture ID and the part ID.

SIZE = width,height

It changes the display size. If it is omitted , or , you specify 0 ,it gets the size from the texture parts.

```
SIZE = 64,32; //64x32
SIZE = ,32;   //Set the width of the texture part width
SIZE = 64;     //Set the height of the texture part height
```

COLOR = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

CAPTION = Caption ID

Set the ID of the string .

Please refer to [here](#) .

CAPTION = “string”

Set the string.

```
CAPTION = "Hello world!";
```

CAPTION_COLOR = R,G,B,A

You can set the caption color .

Set a value in the range of 0~1.

CAPTION_OFFSET = X, Y

It adjust the caption position.

STYLE = Flag0|Flag1|..|Flagn

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

You set the caption position to the center.

Text anchor	Description
TEXT_CENTER	It aligns the text to the center.
TEXT_LEFT	It aligns the text to the left side.
TEXT_RIGHT	It aligns the text to the right side.

Change the font decoration of the caption .

You can change the font decoration of the caption .	Description
TEXT_NORMAL	No decoration
TEXT_BOLD	Bold type
TEXT_DENT	Dent type
TEXT_SHADOW	Shadow Type

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.

10.27 BAR

C#: *CWinCtrlBar*

The control to display a bar .

It can have a caption.

It is possible to receive the onClick event .

```
BAR(Control name) {
    Property 1;
    Property 2;
    :
    :
    Property n
};
```

10.27.1 Example

```
WINDOW(255_000_00001) {
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,-100;
    SIZE = 512,256;
};

$w = 400;
BAR(TEST) {
    ID = 001_000_00020;
    CAPTION = 001_000_00030;
    CAPTION_COLOR = 0,0,0,1;
    STYLE = ANCHOR_LEFTTOP;
    POSITION = 16,-64;
    SIZE = $w,48;
};
```

10.27.2 Property

Default value

```
TEX_ID = "BAR";
STYLE = TEXT_CENTER;
COLOR = 1,1,1,1;
CAPTION_COLOR = 1,1,1,1;
```

ID = Control ID

Define the control ID.

```
ID = 001_000_00010;
```

Note: If you don't set the ID, it generates automatically from the hash value.

POSITION = X, Y

It determine the display position . By the setting of STYLE, you can change the criteria for display position.

```
POSITION = 32,-64;
```

TEX_ID = Texture ID,Part ID

TEX_ID = Part ID

You set the texture ID and the part ID.

SIZE = width,height

It changes the display size. If it is omitted , or , you specify 0 ,it gets the size from the texture parts.

```
SIZE = 64,32; //64x32
SIZE = ,32;   //Set the width of the texture part width
SIZE = 64;     //Set the height of the texture part height
```

COLOR = R,G,B,A

Set the color.

Set R, G, B in the range of 0~2 .

If you set a value in excess of one , each color component can double the brightness.

Set A in the range of 0~1 .

CAPTION = Caption ID

Set the ID of the string .

Please refer to [here](#) .

CAPTION = "string"

Set the string.

```
CAPTION = "Hello world!";
```

CAPTION_COLOR = R,G,B,A

You can set the caption color .

Set a value in the range of 0~1.

CAPTION_OFFSET = X, Y

It adjust the caption position.

SE_ID = SE_ID

To set the ID of the sound played when pressed . 0 is set as the default. This default value means not to play a sound.

STYLE = Flag0|Flag1|..|Flagn

You can set the display anchor of control.

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_LEFTTOP The default control center position is set to BASE_LEFT.
ANCHOR_LEFTTOP	Set the anchor position in the upper left The default control center position is set to BASE_LEFT.
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering The default control center position is set to BASE_LEFT.
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge The default control center position is set to BASE_LEFTBOTTOM.
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral. The default control center position is set to BASE_TOP.
ANCHOR_CENTER	Set the anchor position in the center of the window The default control center position is set to BASE_CENTER.
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral. The default control center position is set to BASE_BOTTOM.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right The default control center position is set to BASE_RIGHTBOTTOM.
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering The default control center position is set to BASE_RIGHT.
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge The default control center position is set to BASE_RIGHTBOTTOM.

You can set the center position of the control.

Base position change flag	Description
BASE_DEFAULT	It varies depending on the anchor flag . If you do not specifically set , this value is set . See the description of each anchor flag
BASE_LEFTTOP	Set the center position to the upper left side.
BASE_LEFT	Set the center position to the left side. and vertical centering.
BASE_LEFTBOTTOM	Set the center position to the bottom left side.
BASE_TOP	Set the center position to the upper side. and horizontal centering
BASE_CENTER	Set the center position to the center.
BASE_BOTTOM	Set the center position to the bottom side. And horizontal centering.
BASE_RIGHTTOP	Set the center position to the upper right side.
BASE_RIGHT	Set the center position to the right side. and horizontal centering
BASE_RIGHTBOTTOM	Set the center position to the bottom right side.

You set the caption position to the center.

Text anchor	Description
TEXT_CENTER	It aligns the text to the center.
TEXT_LEFT	It aligns the text to the left side.
TEXT_RIGHT	It aligns the text to the right side.

Change the font decoration of the caption .

You can change the font decoration of the caption .	Description
TEXT_NORMAL	No decoration
TEXT_BOLD	Bold type
TEXT_DENT	Dent type
TEXT_SHADOW	Shadow Type

Style that limits the function as follows .

Function limit style	Description
HIDE	Hide.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.

Reference:

10.28 CMainSystemBase

The CMainSystemBase adds to the game object . The game object is not deleted even switched the scene

By adding resident managers in this script , you can download related assetbundles.

By performing a scene loading using this script,it waits to finish assetbundles loading.

When using it , create a CMainSystem class and to inherit it .

void **Awake ()**

In Awake method,by calling the AddComponent and addManager, your application can automatically load asset bundles with the necessary initialization.

A typical description method is as follows .

Note: Start () should not be overwritten . It can not be successfully loading .

```

new void Awake() {
    base.Awake();

    if (m_instance != null) {
        Debug.LogError("already exist CMainSystem");
        return;
    }
    m_instance = this;

    // Add Component
    gameObject.AddComponent<CInput>();
    gameObject.AddComponent<CAsetBundleMgr>();
    gameObject.AddComponent<CSpriteFontMgr>();
    gameObject.AddComponent<CTextureResourceMgr>();
    gameObject.AddComponent<CWindowMgr>();
    gameObject.AddComponent<CBgmResourceMgr>();
    gameObject.AddComponent<CSeResourceMgr>();
    gameObject.AddComponent<CSoundEffectMgr>();

    // 初期化の必要なマネージャ .
    addManager(new CMessageDataSheetMgr(utility.getSystemLocale())));
}

```

void **addManager** (IManager *mgr*)

Parameters IManager *mgr* Objects must be initialized by loading the asset bundles

Note: In Awake, it is necessary to AddComponet.

virtual void **initialize** ()

There is a need to override at a inheritance destination. After load assetbundles,load of ILoader, and complete IManager, it is called.

virtual void **OnChangeScene** ()

Override at a inheritance destination . Called when you switch the scene .

bool **changeScene** (string *sScene*, bool *bForceChange* = false)

It is switched to the specified scene .

It is equivalent to the Application.LoadLevel , but it has difference points as follows.

- 1.Wait until the fade of 「fade object」 is finished .
- 2.While assetbundles loading , if you are switching a scene ,this script pends the switching the scene.
- 3.The created windows close automatically.

To the scene immediately before switching (just before the Application.LoadLevel is called), On-ChangeScene is called .

bool **isChangingScene**

Determining whether or not in the scene loading .

It is effective only when you are switching the scene by CMainSystemBase.chageScene.

bool **isInitialized**

It is determined whether it has been completed the initialization.

static int randi ()

You get a random number of integer values.

static int randi (int *iMax*)

You get a random number in the specified range.

static int randi (int *iMin*, int *iMax*)

You get a random number in the specified range.

static float randf ()

You get a random number of floating point.

10.28.1 ILoader interface**bool isLoading ()**

Return bool It is true if loading. It is false if loaded.

These interfaces need to be implemented by a script that inherits MonoBehaviour.

For scripts that implement the ILoader interface, it wait until ILoader.isLoading () becomes false. These scripts should be attached to the game object CMainSystemBase is attached.

Note: In Awake, it is necessary to AddComponet.

10.28.2 IManager interface

These should be added using the CMainSystemBase.addManager in Awake method.

It is an interface required for the objects that must be initialized with the loaded asset bundle.

```
addManager (new CMessageDataSheetMgr (Utility.getSystemLocale ()));
```

bool initialize (CAssetBundle[] aAssetBundles)**Parameters**

- **aAssetBundles** (CAssetBundle[]) – Array of loaded assetbundles

void release ()

Release process that is called when the game object is freed

uint[] getAssetBundleIds ()

Return uint[] array of assetbundle IDs.

It is a function that returns an array of asset bundle IDs that must be loaded.

10.29 CAssetBundleMgr

You need to add this script by AddComponent in “Awake” method of CMainSystem.

This is the manager to cache or load the assetbundles.

When you prepare your own manager , it is unnecessary.

First ,for get the version of each asset , this script downloads the version.unity3d.

if the versions are different ,this script discards old cached data.

To have the following characteristics .

1. Asset bundles are version control.This script does not download assetbundles if same assetbundle downloaded in the past(WWW.LoadFromCacheOrDownload).
2. The loaded asset bundles are cached in memory.
3. It releases the cache at the scene switching .
4. It is possible to put a resident flag in assetbundles. These are not released at the scene switching.
5. Possible to switch the load path in the configuration file on the MS-Windows and the Mac (local files without using the HTTP server can also be loaded)

-
6. The name's format of the assetbundle is *MulID*.unity3d.

10.29.1 CAssetBundleMgr

const int MaxConcurrentLoadNum

This value means the maximum number that this script can download at the same time.

In other words, the number of a LoadFromCacheOrDownload calling can not exceed this maximum value.

The default value is to 1.

const long MaximumAvailableDiskSpace

It sets the maximum cache size of a local disk.

The default value is set to $512 * 1024 * 1024$ (512Mbyte).

static string httpServerPath

It sets the HTTP address.

static string debugPath

It sets the debug path . It is effective only in MS-Windows / Mac environment .This manager preferentially loads assetbundles via this path.

Note: A file that was loaded via the debug path does not perform the cache to the local disk.

int lastError

It has the error number

int errorMessage

It has the error message

CAssetBundle[] loadings

This returns a list of asset bundles currently being downloaded .

This list includes what the manager deferred assetbundle downloads.

int loadNum

It has a number of asset bundle currently being downloaded .

This number includes what the manager deferred assetbundle downloads.

CAssetBundleMgr Instance

It has an instance of a manager

CAssetBundle reference (uint id)

It references a assetbundle. If you have not cache it , the manager downloads it .

If it was already downloaded, it returns the existing CAssetBundle.

Whether CAssetBundle was downloaded, it is possible to determine by CAssetBundle.isLoaded.

if it returns true,the asset is already loaded.

When null is returned , it is not exist.

bool isExist (uint id)

It examines whether the asset bundle exist .

void startPreload()

You need to call this function if you want to download all asset bundles that exist in order to cache on the local disk.

void loadAssetVersion()

Asset version (version.unity3d) to download and again , and updates the cache of asset bundle.

```
void releaseAll()
It releases all asset bundle regardless of the resident / non-resident .
After the release , you must call the loadAssetVersion.
```

10.29.2 CAssetBundle

It can be cast in AssetBundle.

AssetBundle **get** ()
Get the AssetBundle.

uint **id**
It has the ID.

bool **isLoaded**
if it is already loaded, this value sets to true.

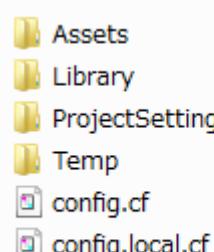
bool **isRemain**
You can see whether it is resident .

How to change a path to load local files.

It is only valid on Windows / Mac.

You make the following files in the same location as the Assets folder.

- config.cf
- config.local.cf



The loading order is config.cf → config.local.cf.

“config.cf” is overwritten by “config.local.cf”.

It is described as follows in config.cf

```
#
# config.cf
#
DEBUG_PATH = ../../unified/AssetBuilder/assetbundles/Windows
```

In the Awake of CMainSystem, set as follows .

```
//=====================================================================
/*!Awake
* @brief      Unity Callback
*/
new void Awake() {
    base.Awake();
    if (m_instance != null) {
        Debug.LogError("already exist CMainSystem");
        return;
}
```

```

    }
    m_instance = this;
    #if !(UNITY_IPHONE || UNITY_ANDROID || UNITY_WEBPLAYER)
        CAssetBundleMgr.debugPath = CConfig.Instance.get ("DEBUG_PATH", ".../Debug/");
    #endif
    gameObject.AddComponent<CAssetBundleMgr> ();
}

```

10.30 About the AssetBundle version.

First, CAssetBundleMgr load the version.unity3d. In this item, it will be explained this version.unity3d.

10.30.1 When is version.unity3d updated?

- [KsTools]->[Export Binary Data]
- [KsTools]->[Export Window Resource]

It automatically updates when you create or update assetbundles.

Tools calls the next function.

```
ExportVersion.export (BuildTarget eTarget);
```

The vresion.unity3d have information as following.

- List of assets bundles
- MD5 of each asset bundle
- Version number of each asset bundle
- Whether asset bundle resident

This function compares the MD5 of assetbundle and the value of version.unity3d.if the value is different,the version.unity3d is updated.

In this case,giving new version number that was generated from the current time.

The manager is able to know following by the version.unity3d.

- You can see in advance whether asset bundles exist
- It's giving the new version to a only file of the update. Thereby, it is possible to download only the updated files. You can use this version number to LoadFromCacheOrDownload.

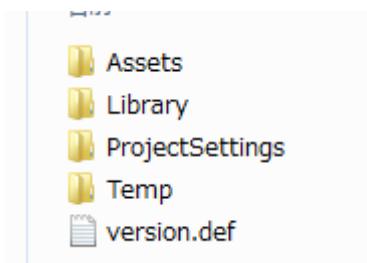
Note: When you want to create an asset bundle in the own, you do not forget that your scripts call to export the version.unity3d in the last.

10.30.2 Resident of asset bundles

The version.unity3d can have flags whether resident assetbundles.

In the same location as the Assets folder , please create a ‘version.def’ file.

```
#Texture for Window Resource
100_000_00001 =      REMAIN:1;
#SoundEffect for 2D
052_000_00000 =     REMAIN:1;
```



```
#SoundEffect for 3D
053_000_00000 = REMAIN:1;
```

Asset bundle specified in this way will be resident assetbundles after download.

10.30.3 a version.txt

This tool visualizes version.txt from version.unity3d for debugging.

This is not necessary on the exporting and runtime.

10.30.4 When a collision occurs in such as version control tool.

Do Take the following steps .

When the asset bundle collides

Resolve the conflict by yourself.

When version.unity3d collide

You should resolve the conflict of assetbundles. You remove the version.unity3d. After that, it will download the version.unity3d that have been uploaded. Finally, you need to output version.unity3d again in the following way in the local environment.

- [KsTools]->[Test]->[Export Version]

10.31 CTextureResourceMgr

You need to add this script by AddComponent in “Awake” method of CMainSystem.

This is intended to manage the texture resources .

It is a container of CTextureResource.

The CTextureResource is downloaded or loaded from resources. This manager returns the cache if it was already loaded.

This window system also load a texture via this manager .

The resource creation method please refer to [here](#) .

Note: To read via asset bundle , you need to register the [*CAssetBundleMgr*](#) .

10.31.1 CTextureResourceMgr

CTextureResource **reference** (uint *id*)

Parameters uint **id** Asset bundle ID

It returns the texture resources. If the resource is not cached ,then it start to download.

When the texture resource exists even if not already downloaded , it will return the CTextureResource.

By examining the CTextureResource.isLoaded, you can know whether was download it.

if it returns true,the asset is already loaded.

If the manager can not download as an assetbundle,it tries to load as resource data.

If it does not exist , the manager returns a null.

CTextureResourceMgr **Instance**

It has an instance of a manager

10.31.2 CTextureResource

This can be cast to the Material.

Material **get** ()

You get the Material .

uint **id**

It has the ID.

bool **isLoaded**

if it is already loaded,this value sets to true.

bool **isRemain**

You can see whether it is resident .

CSpriteDataOne[] **spriteData**

You get the sprite data .

Parts of UV information, patch information , and color information are stored.

10.31.3 CSpriteDataOne

Parts of UV information, patch information , and color information are stored.

uint **m_id**

WinColor **m_color**

e_Patch **m_ePatch**

```
public enum e_Patch {
    None,      //patch none
    H3,        //3 patch in a horizontal direction
    V3,        //3 patch in the vertical direction
    HV9,       //9 patch
};
```

Vector4 **m_uv**

UV texture information are stored .

Vector4[] **m_aUV**

If e_Patch is H3 or V3, it is stored three UV information. If e_Patch is HV9, it is stored nine UV information

10.32 CSoundEffectMgr

This manager have some of the AudioSource in advance. it controls the concurrent sound number limit and their grouping.

The default number of concurrent limits are set as follows:

- 2D Voice: 4 sounds
- 3D Voice: 8 sounds

Also it has become a concurrent number of limitations to the individually set for each group .

How to perform these settings SE , refer to [here](#) .

It manages AudioSource. You do not need to create a AudioSource to play sounds.

10.32.1 How to incorporate the manager

To use CSoundEffectMgr, it is necessary to add managers to the game object as follows.

```
public class CMainSystem : CMainSystemBase {
//=====
/*!Awake
* @brief Unity Callback
*/
new void Awake() {
    base.Awake();

    if (m_instance != null) {
        Debug.LogError("already exist CMainSystem");
        return;
    }
    m_instance = this;

    // Add Component
    gameObject.AddComponent<CInput>();
    gameObject.AddComponent<CAsetBundleMgr>();
    gameObject.AddComponent<CSpriteFontMgr>();
    gameObject.AddComponent<CTextureResourceMgr>();
    gameObject.AddComponent<CWindowMgr>();
    gameObject.AddComponent<CBgmResourceMgr>();
    gameObject.AddComponent<CSeResourceMgr>();
    gameObject.AddComponent<CSoundEffectMgr>();

    addManager(new CMessageDataSheetMgr(Utility.getSystemLocale()));
}
}
```

void **playBgm**(CBgmResource *cBgmResource*)

Parameters

- **CBgmResource** – The loaded BGM resource

void **playBgm**(uint *id*)

param uint id Asset bundle ID

It plays the BGM. If you have not loaded the asset bundle yet, it will automatically start playing after you have finished loading.

To get the CBgmResource directly , you can write as follows.

```
IEnumerator load(uint id) {
    CBgmResource cBgmResource = CBgmResourceMgr.Instance.reference(id);
```

```

if (cBgmResource == null) {
    // error!!
    yield break;
}
while (!cBgmResource.isLoaded()) {
    yield return 0;
}
CSoundEffectMgr.Instance.playBgm(cBgmResource);
};

```

void **play** (CSoundEffect *cSE*)

Parameters

- **cSE** (*CSoundEffect*) – This is the SE objects that can get from CSeResource.

It plays the specified Sound effect as the 2D voice. CSoundEffect is a possible to get in the following ways

```

CSeResource m_cSR;
:
CSoundEffect cSE = m_cSR.find(id);

```

void **play** (uint *mAssetBundle*, uint *id*)

Parameters

- **mAssetBundle** (*uint*) – Asset bundle ID
- **id** (*uint*) – SE ID

It plays the specified Sound effect as the 2D voice.

void **play** (CSoundEffect *cSE*, IEffectEmitter *emitter*)

void **play** (CSoundEffect *cSE*, Transform *trans*)

void **play** (CSoundEffect *cSE*, Vector3 *position*)

Parameters

- **cSE** (*CSoundEffect*) – This is the SE objects that can get from CSeResource.
- **emitter** (*IEffectEmitter*) – Emitter the effect occurs .
- **trans** (*Transform*) – The position at playing the sound.
- **position** (*Vector3*) – The position at playing the sound.

It plays the specified Sound effect as the 3D voice.

CSoundEffectMgr **Instance**

It has an instance of a manager

10.33 CSeResourceMgr

This is in the container to manage the CSeResource.

After you download the asset bundle via CAssetBundleMgr, to manage the voice data that was loaded

CSeResource **reference** (uint *mAssetBundle*)

Parameters

- **mAssetBundle** (*uint*) – Specify the SE resource ID (asset bundle ID).

Returns

It returns a SE resource .

It returns a SE resource. Assetbundles of SE resource were converted by *this way* .

CSoundEffect **find** (uint *mAssetBundle*, uint *mId*)

Parameters

- **mAssetBundle** (*uint*) – Specify the SE resource .
- **id** (*uint*) – Specify the SE ID .

Returns if the asset bundle does not exist or was still being load , it is returned null.

It returns the CSoundEffect.

CSeResourceMgr **Instance**

It has an instance of a manager

10.34 CSeResource

It has some of sound effects in the state that you can play them.

A CSeResource has some of sound effects.

It has a IWinSoundEffect interface. So it is possible to be registered in the CWindowMgr.

If you play the SE, you can use the CSoundEffectMgr .

CSoundEffect **find** (uint *id*)

Parameters

- **id** (*uint*) – Specify a SE ID.

Returns It returns the SE.

Get the SE

void **play** (uint *mSE*)

Parameters

- **id** (*uint*) – Specify the SE ID.

It plays the SE.

bool **isLoaded**

It examines whether the asset bundles are finished loading.

If It returns true, the loading has been completed, and it is ready for use.

10.35 CBgmResourceMgr

This is in the container to manage the CBgmResource.

After you download the asset bundle via CAssetBundleMgr, to manage the BGM data that was loaded

If you want to play the BGM,you play it by the CSoundEffectMgr.

CBgmResource **reference** (uint *mAssetBundle*)

Parameters

- **mAssetBundle** (*uint*) – Specify the BGM resource ID (asset bundle ID).

Returns It returns the BGM resource.

It returns the BGM resource. Assetbundles of BGM resource were converted by *this way* .

CBgmResource **load** (uint *mResource*)

Parameters

- **mResource** (*uint*) – Specify the BGM resource ID (asset bundle ID).

Returns It returns the BGM resource.

Under the Resources folder and place files as follows, it is possible to load them.

```
Assets/Resources/001_000_000000.mp3
Assets/Resources/001_000_000000.intro.mp3
```

CBgmResourceMgr Instance

It has an instance of a manager

10.36 CBgmResource

It has a BGM in the state that you can play it.

CBgmResource has the voice data of the intro part and the loop part .

If you want to play the BGM,you play it by the CSoundEffectMgr.

AudioClip loopClip

Clip of loop part

AudioClip introClip

Clip of intro part

bool isLoaded

It examines whether the asset bundles are finished loading.

If It returns true, the loading has been completed, and it is ready for use.

10.37 CMessageDataSheetMgr

It is an object that is used to manage the string resources.

CMessageDataSheet corresponds to the an Excel sheet .

CMessageDataSheetMgr is the container of CMessageDataSheet.

Just CMessageDataSheetMgr corresponds to an Excel file .

10.37.1 How to use in your application

You need to add this script by addManager in “ Awake ” method of CMainSystem.

```
public class CMainSystem : CMainSystemBase {
//=====
/*!Awake
* @brief Unity Callback
*/
new void Awake() {
    base.Awake();

    if (m_instance != null) {
        Debug.LogError("already exist CMainSystem");
        return;
    }
    m_instance = this;

    // Add Component
```

```

gameObject.AddComponent<CInput>();
gameObject.AddComponent<CAAssetBundleMgr>();
gameObject.AddComponent<CSpriteFontMgr>();
gameObject.AddComponent<CTextureResourceMgr>();
gameObject.AddComponent<CWindowMgr>();
gameObject.AddComponent<CBgmResourceMgr>();
gameObject.AddComponent<CSeResourceMgr>();
gameObject.AddComponent<CSoundEffectMgr>();

addManager (new CMessageDataSheetMgr(utility.getSystemLocale()));
}

```

Also, if you want to associate it with the caption of CWindowMgr, you need to describe as follows in the initialize of CMainSystem.

```

public class CMainSystem : CMainSystemBase {
//=====
/*! Initialize
* @brief      initialize
*/
override protected void initialize() {
    base.initialize();

//-----
// WindowMgr initialize.
//-----
    CWindowMgr      cWindowMgr = CWindowMgr.Instance;
    // initialize caption interface
    cWindowMgr.captiondata = CMessageDataSheetMgr.Instance.find(new FiveCC("WNDW"));
}

```

Since CMessageDataSheetMgr has a IWinCaptionData interface , this is possible .

10.37.2 CMessageDataSheetMgr

CMessageDataSheetMgr is a container of CMessageDataSheet.

CMessageDataSheet corresponds to the an Excel sheet .

In other words,CMessageDataSheetMgr is the container of your Excel sheets.

However,the assetbundle of string resource has only one locale.(it does not have multiple locale data).

void **initialize** (byte[] *buffer*)

Parameters

- **buffer** (byte[]) – The loaded binary data

It specifies the binary data generated from Excel.

CMessageDataSheet **find** (uint *sheetname*)

Parameters uint type Sheet be named in *FiveCC* .

Reference a sheet .

```
CMessageDataSheet cSheet = cMessageDataSheetMgr.find(new FiveCC("WNDW"));
```

e_Locale **eLocale**

Get the currently set locale .

CMessageDataSheetMgr **Instance**

Get an instance of CMessageDataSheetMgr.

10.37.3 CMessageDataSheet

It corresponds to the one sheet on an Excel file.

string **find** (uint *id*)

Parameters

- **id** (*uint*) – It gets the string resources by specifying the *MulID* .

string **format** (uint *id*, params object[] *args*)

Parameters

- **id** (*uint*) – It gets the string resources by specifying the *MulID* .
- **object[] args** (*params*) – Variable argument

Return the generated string data along the format.

```
string str = cMessageDataSheetMgr.format(new MulId(0, 0, 1), "Test", n, value);
```

10.38 CWindowMgr

You need to add this script by AddComponent in “Awake” method of CMainSystem.

This script manages windows.

All windows are created by this manager.

10.38.1 CWindowMgr

bool **initialize** (CAssetBundle[] *aAB*)

Expand and Initialize the loaded window resources. If you have registered CAssetBundleMgr, this manager will automatically download the window resources from the server. After the completion of loading the window asset bundle, it is passed to this function.

bool **load** (byte[] *buffer*)

It is initialized by the byte data of loaded window resource .

bool **load** (string *assetname*)

Parameters **string assetname** A resource name

Load the window resources of argument name .

An asset name is *assetname* + ”. asset ”.

WindowScript **create**<WindowScript> (uint *id*, CWindowBase *cParent* = null)

Parameters

- **WindowScript** – Class name of the create the window
- **uint id** – Window ID
- **CWindowBase cParet** – Parent window

Create the window.

WindowScript **find**<WindowScript> (uint *id*)

Parameters

- **WindowScript** – Window class name
- **uint id** – Window ID

Search the window. If not yet been create the window , it returns null.

void **bringToTop** (CWindowBase *cWindow*)

Parameters CWindowBase *cWindow* window

It raises the priority of the target window.

string **getCaption** (uint *mCaptionId*)

Parameters uint *mCaptionId* Caption ID

Get the string from the caption ID.

void **play** (uint *mSE*)

Parameters uint *mSE* SE ID

Play the sound effect of SE ID.

uint **clickSE**

Set the standard playing sound when you touch.

uint **scrollSE**

Set the standard playing sound when you scroll through the list box .

static CWindowMgr **Instance**

Get an instatnce of window manager

10.39 CWindowBase

10.39.1 How to create a window class

If you succeed in compiling a window script , it generate the window binary file and the base file for using the window.

For example

```
#include "wr.h"
WINDOW(TEST) {
    ID = 255_000_00001;
    STYLE = WINDOW_STYLE_NOTITLEBAR|WINDOW_STYLE_ANCHOR_CENTER;
    POSITION = 0,-100;
    SIZE = 512,256;
};

TEXTBOX(Message) {
    ID = 000_000_00010;
    POSITION = 100,-80;
    CAPTION = 000_111_00020;
    EDIT = 255,2;
    SIZE = 380, 64;
};
```

CTestBase.cs is automatically generated after the compilation of **CTest.wra** .

```
public class TESTBase : CWindowBase {
    public const uint windowId = 4278190081; // 255_000_00001
    static public TEST create(CWindowBase cParent = null) {
        return CWindowMgr.Instance.create<TEST>(windowId, cParent);
    }
    public const uint TEXTBOX_Message = 10; // 000_000_00010
};
```

You make a window class that inherits **CTestBase** .

```
public class TEST : TestBase {
    :
}
```

Then ,when actually create the window, as follows .

```
TEST test = TEST.create();
```

10.39.2 The callback that reach to each window object

A window is composed by many controls.

If you perform operations as touch and drag to controls, the callback function in this window is called.

It enumerates callback functions that a window can receive.

Inherit the class that is generated when you compile the script window. Then,necessary,you need to override callback functions.

virtual void onCreate()

It is called immediately after the initialization of the window has been completed .

virtual void onUpdate()

It is called before the update of each control.

virtual void onAfterUpdate()

It called after the update of each control.

virtual void onPreRender(CWindowRenderer cRenderer)

It is called before rendering the control.

virtual void onRender(CWindowRenderer cRenderer)

It is called after the control of the rendering is finished.

virtual bool onClose(int iCloseInfo)

It is called when you close the window.

true: actually close the window .

false: cancel to close the window .

virtual void onClick(CWinCtrlBase cCtrl)

Pressed the control

virtual void onHold(CWinCtrlBase cCtrl)

Pressed and Holded the control

virtual void onClickEnter(CWinCtrlBase cCtrl)

Determine the input of the edit box.

virtual void onClick(CWinCtrlBase cCtrl, CRichTextOne cText)

Pressed a part of the rich text.

virtual void onDrag(CWinCtrlBase cCtrl)

Dragged the control

virtual void onDrag(CWinCtrlBase cCtrl, Vector2 pos)

Dragged and moved the control

virtual void onDragRender(CWinCtrlBase cCtrl, CWindowRenderer renderer, Vector2 pos)

If you set true to this value,this window become not to respond.

virtual void onDrop(CWinCtrlBase cCtrl, CWindowBase cDragWindow, CWinCtrlBase cDragCtrl)

It is called when the control being dragged has been released on the control of specific window.

virtual void onDropGround(CWinCtrlBase cCtrl, CWindowBase cDragWindow, CWinCtrlBase cDragCtrl)

if you drop the dragged control on the nothing place,it is called.

virtual void onOK (int *msgBoxWinID*)

Pressed OK button in the message window. This callback will reach the parent window of the message window.

virtual void onNext (int *msgBoxWinID*)

Pressed Next button in the message window. This callback will reach the parent window of the message window.

virtual void onYes (int *msgBoxWinID*)

Pressed Yes button in the message window. This callback will reach the parent window of the message window.

virtual void onNo (int *msgBoxWinID*)

Pressed No button in the message window. This callback will reach the parent window of the message window.

virtual void onCancel1 (int *msgBoxWinID*)

Pressed Cancel button in the message window. This callback will reach the parent window of the message window.

virtual void onScroll1 (CWinCtrlBase *cCtrl*, Vector2 *velocity*)

Scroll by dragging a list box or container.

virtual void onRecreatedRenderTexture (uint *mRenderTextureId*)

This function is called when lost Render Texture by some event. However , it is not necessary to override it because the automatic return to the system side is performed.

10.39.3 Window property/method

I explain the accessible methods and properties of CWindowBase.

void close (int *iCloseInfo* = 0)

Call when you want to close the window. *iCloseInfo* is passed as a parameter of onClose.

WinCtrl find<WinCtrl> (uint *id*)

Look for a control with the ID.

CWinCtrlBase find (uint *id*)

Look for a control with the ID.

uint id

Get/Set the window ID

CWindowBase child

Return the child window

CWindowBase parent

Return the parent window

e_LayerId layer

Get / Set the rendering layer

int priority

Get / Set the drawing priority .

uint texid

Return the default texture ID of the window .

float x

float y

Get / Set the window position .

Vector2 position

Get / Set the window position .

Transform **transform**

Get the transform

Vector2 **scale**

Get/Set x,y scales

e_Anchor **anchor**

set the anchor for display position.

float **width**

float **height**

Get/Set the window size

Vector2 **size**

Get/Set the window size

float **screenWidth**

float **screenHeight**

Get/Set the screen size

string **caption**

Get / Set the title bar caption of the window.

WinColor **captionColor**

Get / Set the title bar caption color of the window.

Vector2 **captionOffset**

Get / Set the title bar caption offset of the window.

WinColor **color**

Set the window color. All controls in the window is affected.

e_WinStyle **style**

Get the window style.

e_WinStyle **priorityStyle**

Get/Set the window priority style.

bool **hide**

true Hide the window display.

false Display the window.

bool **disable**

true It deprives functions of the window, and the window darken.

false Display the window

When depriving the window function, it will not respond even touching.

bool **isClose**

This value is true among the window play the animation of closing.

bool **isLoaded**

This value is true among the texture loading.

bool **isFade**

This value is true among the fade state of window is in or out.

10.39.4 Other overridable methods

CWindowBase has overridable methods other than callback. You need to override when you want to customize the fade animation when creating the window or closing the window.

protected virtual void startFade (e_FadeState eState)

It is called when the fade began. You can distinguish fade state whether IN or OUT by the function argument.

e_FadeState.Open Fade in

e_FadeState.Close Fade out

protected virtual bool doFade (e_FadeState eState)

フェード中に呼ばれます。

true if you want to continue the fade state.

false if you want to finish the fade state.

if you want to finish the fade state, you need to return false.

In addition, even if this function argument is the e_FadeState.None,you need to return false.

e_FadeState.None Do not a fade

e_FadeState.Open Among a fade-in

e_FadeState.Close Among a fade-out

10.40 CWinCtrlBase

All controls have inherited CWinCtrlBase.

CWinCtrlBase has common methods / properties of each control.

Control that inherits CWinCtrlBase is also one that has its own properties. Please refer to the description of each control about it.

10.40.1 About contents

By *getContents()*, you can access controls that are defined by the CONTENTS property.

The corresponding control: *CHECKBOX* , *RADIO* , *CONTAINER*

```
// Get the CONTAINER(Map)
CWinCtrlContainer      ctnMap = find<CWinCtrlContainer>(CONTAINER_Map);
CWinContents contents = ctnMap.getContents();
// Get the BUTTON(A) in contents
CWinCtrlButton btnA = contents.find<CWinCtrlButton>(BUTTON_A);
```

By *CWindowBase.find* and *CWinContents.find* , you can get the specific control on the contents.

```
CWinCtrlButton btnA = find<CWinCtrlButton>(BUTTON_A);
```

10.40.2 Contents list

List box have a lot of contents. To add a row in the list box, you need to duplicate the *contents* as a template. Therefore, the control with the same control ID exists on the contents list. To get a control on the contents list, it is necessary to identify the contents from the contents list.

In this case, you use the *getContentsFromIndex(int index)* .

It shows a typical code for access to the control in the list box.

```
// get the listbox control
CWinCtrlListbox      lbFriends = find<CWinCtrlListbox>(LISTBOX_Friends);

// set the listbox contents number
lbFriends.resize(10);
```

```
// update each contents in listbox(update 10 contents).
for (int i = 0; i < lbFriends.Count; ++i) {
    // get the one column(= contents) from listbox
    CWinContents contents = lbFriends.getContentsFromIndex(i);

    CWInCtrlIcon icon = contents.find<CWInCtrlIcon>(ICON_Avatar);
```

10.40.3 The main properties / methods

For the contents

CWinContents **getContents ()**

Return CWinContents Get the *contents* that is defined by *CONTENTS* property.

For the contents list

int **count**

Get a number of contents list .

void **resize (int count)**

Parameters int count The number of contents list

Set the number of contents list .

When the number of the contents list increased, a necessary number are copied from the contents as a template.

It does not operate an element in the list of contents which existed already.

CWinContents **insert (int pos)**

Parameters int pos After contents generation , you specify a position that you want to insert it on the contents list.

Return CWinContents *Contents* on the contents list.

Copy the contents as a template, and insert it to the specified location.

CWinContents **getContentsFromIndex (int index)**

Parameters int index Contents index that you want to get it on the contents list

Return CWinContents *Contents* on the contents list.

Get the contents using the index from the contents list.

int **getContentsIndex (CWinCtrlBase ctrl)**

It returns the index that the control on the contents list is included in. if the control is not included on the contents list, It returns -1.

int **getContentsIndex (CWinContents contents)**

It returns the index that the contents on the contents list is included in. if the contents is not included on the contents list, It returns -1.

Others

string **ToSString ()**

It generates a string by the type and ID of the control.

uint **id**

the ID of the control.

e_WinCtrlKind kind
the kind of control.

int priority
the priority of the control.

Vector3 position
set the control position.

Vector2 absPosition
Get the window relative position of the control.

Vector2 screenPosition
Get the absolute position of the control

float width

float height
Get / Set the size of the control

Vector2 size
Get / Set the size of the control

Vector2 contentsSize
Get / Set the size of the contents

WinColor getColor (int index)
Set between index = 0 ~ 7. Set the color. You can define up to eight texture parts . Therefore, you can define the eight color at the maximum.

void setColor (int index, WinColor color)
Set between index = 0 ~ 7. Set the color. You can define up to eight texture parts . Therefore, you can define the eight color at the maximum.

WinColor color
Get / Set the color of the texture part number 0 .

WinColor color1

WinColor color2

WinColor color3

WinColor color4

WinColor color5

WinColor color6

WinColor color7
Get / Set the color of the texture part number 1 (~ 7).

void setTextureId (int index, uint texid)
Set between index = 0 ~ 7. Set the texture ID. You can define up to eight texture parts . Therefore, you can define the eight texture at the maximum.

WinColor getTextureId (int index)
Set between index = 0 ~ 7. Get the texture ID. You can define up to eight texture parts . Therefore, you can define the eight texture at the maximum.

uint texId

uint texId1
Get / Set a texture ID.

void setPartId (int index, uint partId)
Set between index = 0 ~ 7. Set the texture part ID. You can define up to eight texture parts . Therefore, you can define the eight texture part ID at the maximum.

uint getPartId (int index)

Set between index = 0 ~ 7. Get the texture part ID. You can define up to eight texture parts . Therefore, you can define the eight texture part ID at the maximum.

void setTextureSize (int index, Vector2 size)

Set between index = 0 ~ 7. Set the texture part size. You can define up to eight texture parts . Therefore, you can define the eight texture part size at the maximum. When index = 0, it becomes the equivalent of accessing (width, height).

uint parts

uint partId1

uint partId2

uint partId3

uint partId4

uint partId5

uint partId6

uint partId7

Get / Set the texture part ID.

void setTextureOffset (int index, Vector2 offset)

Set between index = 0 ~ 7. Set the display offset of texture parts. You can define up to eight texture parts . Therefore, you can define the eight display offsets at the maximum.

Vector2 getTextureOffset (int index)

Set between index = 0 ~ 7. Get the display offset of texture parts. You can define up to eight texture parts . Therefore, you can define the eight display offsets at the maximum.

e_Anchor anchor

Get/Set the anchor to determine the origin .

e_Anchor baseAnchor

Set/Get the base anchor to determine to where the center of the control .

e_Anchor textAnchor

Set/Get the text anchor to decide to where the center of the caption .

uint fontKind

Get / Set the type of font.

string caption

Get / Set the caption.

uint captionId

Get / Set caption ID. However, the setting does not affect caption..

WinColor captionColor

Get / Set the caption color.

Vector2 captionOffset

Get / Set the display offset of caption.

float lineSpace

Set the space between the contents. In addition, in a RichText, it becomes the width of the space between the lines.

uint SEId

Get / Set the ID of the SE assigned to the control .

CWinContents parent

Get the contents that this control is included in.

CWinCtrlBase[] groups

Get the group with the control .

CWindowBase window

Get the window that contains the control.

e_WinCtrlStyle style

Get the control style flag.

bool hide

true: Hide the control.

false: Display the control

bool disable

- true: Inactivate control and darken.

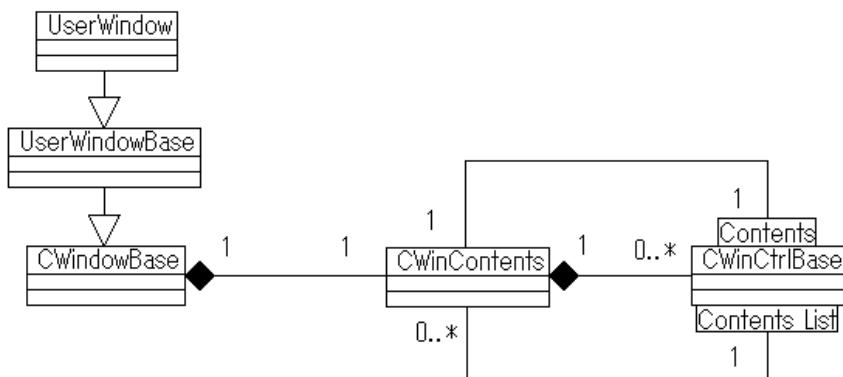
- false: Activate control.

bool nohit

- true: Take away the functions of the control(does not become dark unlike disable).

- false: Activate control.

10.41 CWinContents



CWinContents has a role as a container for holding the various controls .

CWinContents is used in the following part.

- CWindowBase has a CWinContents. Thereby , a window can have some controls.
- CWinCtrlBase has a CWinContents. (This window system calls it the contents. Please refer to [here](#)).
- CWinCtrlBase has it in a list form. This window system calls it the contents list.

The first two are defined structure by the *window script*.

The contents list(third) has a lot of contents by the list form.

10.41.1 Functions/Methods

WinCtrl **find**<WinCtrl> (uint *ctrlId*)

Parameters uint **ctrlId** Control ID

It finds the control of the ID. If it does not exist in the contents , this function recursively searches into the contents of the control. However , this function does not search into the contents list. Thus , control is determined always unique.

int **getIndex** (uint *ctrlId*)

Parameters uint **ctrlId** Control ID

It looks for whether the specified control ID is stored in what number. If not found , it returns -1. This function does not search recursively unlike the 「find」 .

you can use a index value as an argument of [N]. Thereby enabling high-speed access.

CWinCtrlBase **this[N]**

You can use the index value obtained by 「*getIndex*」 .

int **count**

It returns the number of controls in the contents.

ClipRect::State **clipState**

It uses for the contents included within the contents list. (ex: *LISTBOX* , *LISTBOXEX*).

ClipRect.State.Inside Completely within the region

ClipRect.State.Clipped Clipping is necessary.

ClipRect.State.Outside Rendering is unnecessary (completely outside the region)

float **width**

Get the width of the contents. It is automatically calculated from the size of the control.

float **height**

Get the height of the contents. It is automatically calculated from the size of the control.

Vector2 **size**

Get the size of the contents. It is automatically calculated from the size of the control.

Vector3 **position**

Display position of contents

CWinCtrlBase **parent**

Get a parent of this contents.

if the parent of this contents is *CWindowBase* ,it returns a null.

10.42 CWinCtrlText

Script: *TEXT*

It is a control for displaying text .

If you want to change the text color,insert line feeds,or change the font type, please use the *CWinCtrlRichText* .

10.42.1 Callback reaching the CWindowBase

None

10.42.2 Functions / Properties

None

10.42.3 Example of use

```
CWinCtrlText ctrlName = find<CWinCtrlText>(TEXT_Name);
ctrlName.caption = "Mike";
```

10.43 CWinCtrlRichText

Script: *RICHTEXT*

It is a control for displaying rich text.

It is possible to change the text color , insert line feeds, or change the font type

10.43.1 Callback reaching the CWindowBase

void **onClick** (CWinCtrlRichText *cCtrl*, CRichTextOne *cText*)

Parameters

- **CWinCtrlRichText cCtrl** – A touched rich text control.
- **CRichTextOne cText** – A touched part of rich text control.

This callback is called when you touch the rich text part of the texture command ([\t](#)) or the window command([\w](#)) . refer to [here](#) .

void **onHold** (CWinCtrlBase *cCtrl*)

Parameters CWinCtrlBase cCtrl

Control that was pressed and hold

void **onDrag** (CWinCtrlBase *cCtrl*, Vector2 *pos*, Vector2 *dragVelocity*)

Parameters

- **CWinCtrlBase cCtrl** – The dragged control
- **Vector2 pos** – Drag position , or the position of the mouse cursor
- **Vector2 dragVelocity** – Displacement of the touch position

10.43.2 Functions / Properties

10.43.3 CRichTextOne

This is the argument passed to the click event. By examining the member, it is possible to determine where clicked .

CRichTextOne::e_Cmd **cmd**
e_Cmd.String

String : This part click event does not occur.

e_Cmd.Window

Window: part of the `\w command` is clicked .

The following property is enabled .

`userID`

`windowId`

`e_Cmd.Texture`

Texture: part of the `\t command` is clicked .

The following property is enabled .

`userID`

`texId`

`partId`

`uint userId`

User ID is stored. If you omit the user ID, it has 0 .

`uint windowId`

You can get the window ID specified in the `\w command`.

`uint texId`

You can get the texture ID specified in the `\t command`.

`uint partId`

You can get the parts ID set by the `\t command` .

10.44 CWinCtrlLog

Script: `LOG`

This is the control to display a chat log.

It has contents of the same number as the number of rows.

Unlike the list box , the maximum number of contents can be set. If it is added more than the maximum number , the contents is circulated and it is reused(The oldest log is overwritten).

`LOGTEXT` and other kind of controls can also be included as contents.

10.44.1 Callback reaching the CWindowBase

`void onDrag (CWinCtrlBase cCtrl, Vector2 pos, Vector2 dragVelocity)`

Parameters

- `CWinCtrlBase cCtrl` – The dragged control
- `Vector2 pos` – Drag position , or the position of the mouse cursor
- `Vector2 dragVelocity` – Displacement of the touch position

Called when you scroll . Even if you lock the scroll , this callback is called .

10.44.2 Functions / Properties

`void resize (int count)`

Parameters `int count` The log size

It is equivalent to the number of contents in the Listbox. In the other words,it means the maximum number of logs.

int count

It is the number of content in the contents list in the Listbox. In the other words,it means the maximum number of logs.

Never be greater than the value set by the resize.

void clearLog()

Empty all the logs that are currently held.

CWinCtrlLogText add (string text, WinColor color)**Parameters**

- **string text** – Add string
- **WinColor color** – String color

Return CWinCtrlLogText Log text control that set the string

At first, it adds contents internally to add a log(or reuse a log).

This function searches [CWinCtrlLogText](#) from this contents. Then this function sets a string and a color to CWinCtrlLogText.

You can get the CWinCtrlLogText as return value.

```
CWinCtrlLogText ltChat = lgChat.add(strSentence,WinColor.white);
//Access to content that contains the added LogText.
CWinContents contents = ltChat.parent;
```

Vector2 viewsize

the size of list box. This is the same as the size property.

Vector2 screensize

It is a virtual screen size of the list box. It becomes the sum total of all of the height of the contents list.

Vector2 offset

It is a place that is a point of virtual screen. Move at once If you set the value. If you'd like to scroll it smoothly , you can use [setSmoothOffset](#) .

bool isSwipe

Get the state whether is swiped.

void setSmoothOffset (Vector2 offset, float spd)**Parameters**

- **Vector2 offset** – offset
- **float spd** – Speed

Vector2 smoothOffset

Same behavior as [setSmoothOffset](#) with 0.25 speed

bool isSmoothScrolled

Get the state whether is scrolled by setSmoothOffset.

Vector2 getContentsOffset (int index, e_Anchor eAnchor)**Parameters**

- **int index** – The index of contents offset
- **e_Anchor eAnchor** – The position of contents is specified with an anchor.

You can get the offset which is displayed along the anchor.

If the contents of the list box are arranged in a vertical direction

e_Anchor	Location
Bottom,LeftBottom,RightBottom	Offset value for the contents to be aligned with the bottom of the view.
Top,LeftTop,RightTop	Offset value for the contents to be aligned with the top of the view
Center,Left,Right	Offset value for the contents to be aligned with the center of the view

If the contents of the list box are arranged in a horizontal direction

e_Anchor	Location
Left,LeftBottom,LeftTop	Offset value for the contents to be aligned with the left of the view
Right,RightBottom,RightTop	Offset value for the contents to be aligned with the right of the view
Center,Top,Bottom	Offset value for the contents to be aligned with the center of the view

10.44.3 Example of use

```
CWinCtrlLog    lgChat = find<CWinCtrlLog>(LOG_Chat);

lgChat.resize(10);

string strSentence;
CWinCtrlLogText ltChat = lgChat.add(strSentence, WinColor.white);

CWinContents contents = ltChat.parent;
```

10.44.4 How to clear log.

```
//log clear
lgChat.clearLog();
```

10.45 CWinCtrlLogText

Script: *LOGTEXT*

This is a control for having the sentence of the *chat log*.

Note: The *LOG's contents* always requires a LOGTEXT.

10.45.1 Callback reaching the CWindowBase

None

10.45.2 Functions / Properties

None

10.45.3 Example of use

refer to the *CWinCtrlLog example*

10.46 CWinCtrlEditbox

Script: *EDITBOX*

It is a text editable control.

You can set the maximum rows and the maximum string size.

10.46.1 Callback reaching the CWindowBase

void onClick (CWinCtrlBase cCtrl)

Parameters CWinCtrlBase cCtrl The touched control

void onDrag (CWinCtrlBase cCtrl, Vector2 pos, Vector2 dragVelocity)

Parameters

- **CWinCtrlBase cCtrl** – The dragged control
- **Vector2 pos** – Drag position , or the position of the mouse cursor
- **Vector2 dragVelocity** – Displacement of the touch position

void onClickEnter (CWinCtrlBase cCtrl)

Parameters CWinCtrlBase cCtrl Control

It is called when a input string has been determined . The input string is stored in the caption.

10.46.2 Example of use

```
// コントロールを取得する方法.
CWinCtrlEditbox ebSentence = find<CWinCtrlEditbox>(EDITBOX_Sentence);

// 入力された文字列を取得する方法.
string strSentence = ebSentence.caption;

// 入力が完了したときに呼び出される.
override protected void onClickEnter(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
        case EDITBOX_Sentence:
            //押されたときの処理
            break;
    }
}
```

10.47 CWinCtrlTextbox

Script: *TEXTBOX*

This is a edit box,but it can't be edited.

If the caption does not fit to the control size , it is possible to scroll.

10.47.1 Callback reaching the CWindowBase

void onClick (CWinCtrlBase cCtrl)

Parameters CWinCtrlBase cCtrl The touched control

```
void onDrag (CWinCtrlBase cCtrl, Vector2 pos, Vector2 dragVelocity)
```

Parameters

- **CWinCtrlBase cCtrl** – The dragged control
- **Vector2 pos** – Drag position , or the position of the mouse cursor
- **Vector2 dragVelocity** – Displacement of the touch position

10.47.2 Functions / Properties

None

10.47.3 Example of use

```
// コントロールを取得する方法.
CWinCtrlTextbox tbSentence = find<CWinCtrlTextbox>(TEXTBOX_Sentence);

// 文字列を取得する方法.
string strSentence = tbSentence.caption;
```

10.48 CWinCtrlButton

Script: *BUTTON*

This is a button control .

It is possible to display some of the badge and the caption.

It can have seven badges at the maximum.

10.48.1 Callback reaching the CWindowBase

```
void onClick (CWinCtrlBase cCtrl)
```

Parameters CWinCtrlBase *cCtrl* The touched control

```
void onHold (CWinCtrlBase cCtrl)
```

Parameters CWinCtrlBase *cCtrl* Control that was pressed and hold

```
void onDrag (CWinCtrlBase cCtrl, Vector2 pos, Vector2 dragVelocity)
```

Parameters

- **CWinCtrlBase cCtrl** – The dragged control
- **Vector2 pos** – Drag position , or the position of the mouse cursor
- **Vector2 dragVelocity** – Displacement of the touch position

10.48.2 Example of use

```
// コントロールを取得する方法.
CWinCtrlButton ctrlStart = find<CWinCtrlButton>(BUTTON_Start);

// 押されたときのコールバック
override protected void onClick(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
```

```

    case BUTTON_Start:
        //押されたときの処理
        break;
}
// 長押しされたときのコールバック
override protected void onHold(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
        case BUTTON_Start:
            //押されたときの処理
            break;
    }
}

```

10.49 CWinCtrlRadio

Script: *RADIO*

Radio buttons can be grouped radio buttons with each other. When one of the radio button is turned on, the other radio button automatically turns off.

It has the *CONTENTS* as *CHECKBOX*. So it is possible to behave as a tab button.

10.49.1 Callback reaching the CWindowBase

void **onClick** (CWinCtrlBase *cCtrl*)

 Parameters CWinCtrlBase **cCtrl** The touched control

void **onHold** (CWinCtrlBase *cCtrl*)

 Parameters CWinCtrlBase **cCtrl** Control that was pressed and hold

void **onDrag** (CWinCtrlBase *cCtrl*, Vector2 *pos*, Vector2 *dragVelocity*)

 Parameters

- **CWinCtrlBase cCtrl** – The dragged control
- **Vector2 pos** – Drag position , or the position of the mouse cursor
- **Vector2 dragVelocity** – Displacement of the touch position

10.49.2 Functions / Properties

bool **check**

Get/Set the state of a radio button.

true: ON state. Grouped radio buttons turn off. If it has a CONTENTS,it sets CONTENTS to active.

false: OFF state. If it has a CONTENTS,it sets CONTENTS to inactive and hide.

10.49.3 Example of use

```

// コントロールを取得する方法.
CWinCtrlRadio rdSelect = find<CWinCtrlRadio>(RADIO_Select);

if (rdSelect.check) {
    ON の状態
} else {
    OFF の状態
}

```

```

}

// 振されたときのコールバック
override protected void onClick(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
        case RADIO_Select:
            //押されたときの処理
            break;
    }
}
// 長押しされたときのコールバック
override protected void onHold(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
        case RADIO_Select:
            //長押しされたときの処理
            break;
    }
}

```

10.50 CWinCtrlCheckbox

Script: *CHECKBOX*

This is a checkbox button control .

It can display seven badges in the same way as buttons .

It is possible to have *contents*. It associates the controls to be activated when the button is on.

10.50.1 Callback reaching the CWindowBase

void **onClick** (CWinCtrlBase *cCtrl*)

Parameters CWinCtrlBase *cCtrl* The touched control

void **onHold** (CWinCtrlBase *cCtrl*)

Parameters CWinCtrlBase *cCtrl* Control that was pressed and hold

void **onDrag** (CWinCtrlBase *cCtrl*, Vector2 *pos*, Vector2 *dragVelocity*)

Parameters

- **CWinCtrlBase cCtrl** – The dragged control
- **Vector2 pos** – Drag position , or the position of the mouse cursor
- **Vector2 dragVelocity** – Displacement of the touch position

10.50.2 Functions / Properties

bool **check**

Change the state of a check box button.

true: ON state. Grouped radio buttons turn off. If it has a CONTENTS,it sets CONTENTS to active.

false: OFF state. If it has a CONTENTS,it sets CONTENTS to inactive and hide.

10.50.3 Example of use

```
// コントロールを取得する方法.
CWinCtrlCheckbox cbSelect = find<CWinCtrlCheckbox>(CHECKBOX_Select);

if (cbSelect.check) {
    ON の状態
} else {
    OFF の状態
}
// コンテンツ内から、BUTTON(A) を取得する
// コンテナ内のコンテンツを次の通り、find でアクセス可能です
CWinCtrlButton btnA = find<CWinCtrlButton>(BUTTON_A);

// 押されたときのコールバック
override protected void onClick(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
        case CHECKBOX_Select:
            //押されたときの処理
            break;
    }
}
// 長押しされたときのコールバック
override protected void onHold(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
        case CHECKBOX_Select:
            //押されたときの処理
            break;
    }
}
```

10.51 CWinCtrlTexture

Script: *TEXTURE*

This is a control for displaying a texture part .

It is possible to display eight texture parts.

10.51.1 Callback reaching the CWindowBase

void **onClick** (CWinCtrlBase *cCtrl*)

Parameters CWinCtrlBase *cCtrl* The touched control

void **onHold** (CWinCtrlBase *cCtrl*)

Parameters CWinCtrlBase *cCtrl* Control that was pressed and hold

void **onDrag** (CWinCtrlBase *cCtrl*, Vector2 *pos*, Vector2 *dragVelocity*)

Parameters

- **CWinCtrlBase cCtrl** – The dragged control
- **Vector2 pos** – Drag position , or the position of the mouse cursor
- **Vector2 dragVelocity** – Displacement of the touch position

10.51.2 Functions / Properties

None

10.51.3 Example of use

```
// コントロールを取得する方法.
CWinCtrlTexture texAttribute = find<CWinCtrlTexture>(TEXTURE_Attribute);

// 押されたときのコールバック
override protected void onClick(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
        case TEXTURE_Attribute:
            //押されたときの処理
            break;
    }
}

// 長押しされたときのコールバック
override protected void onHold(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
        case TEXTURE_Attribute:
            //押されたときの処理
            break;
    }
}
```

10.52 CWinCtrlLine

Script: *LINE*

This is the control to draw a line segment.

10.52.1 Callback reaching the CWindowBase

None

10.52.2 Example of use

```
// コントロールを取得する方法.
CWinCtrlLine lnTest = find<CWinCtrlLine>(LINE_TEST);
```

10.53 CWinCtrlRender

Script: *RENDER*

It is a control for the Render Texture.

When the control is generated, generates the camera together.

Objects rendered on the camera is drawn.

Note: if you have loaded resources, you need to delete them when you close window.

If you forget to delete them, your application memory will leak.

10.53.1 Callback reaching the CWindowBase

void **onClick** (CWinCtrlBase *cCtrl*)

Parameters **CWinCtrlBase cCtrl** The touched control

```
void onHold (CWinCtrlBase cCtrl)
```

Parameters **CWinCtrlBase cCtrl** Control that was pressed and hold

```
void onDrag (CWinCtrlBase cCtrl, Vector2 pos, Vector2 dragVelocity)
```

Parameters

- **CWinCtrlBase cCtrl** – The dragged control
- **Vector2 pos** – Drag position , or the position of the mouse cursor
- **Vector2 dragVelocity** – Displacement of the touch position

10.53.2 Functions / Properties

Camera **camera**

This function returns the camera to render texture is assigned. if you want to customize a position or an angle of the camera, please access it.

10.53.3 Example of use

```
// コントロールを取得する方法.
public override void onCreate() {
    CWinCtrlRender      rdAvatar = find<CWinCtrlRender>(RENDER_AVATAR);

    // カメラを取得する.
    Camera      camera = rdAvatar.camera;

    // 描画レイヤーを選択.
    camera.cullingMask = (int) e_Layer.OwnPlayer;

    // カメラの位置と角度を決定.
    camera.transform.position = new Vector3(0f, 0.85f, 0f);
    camera.transform.rotation = Quaternion.Euler(new Vector3(0f, 180f, 0f));

    GameObject goAvatar = LoadAvatar();
    // レンダリング対象のゲームオブジェクトのレイヤーを設定 (カメラと同じにする).
    Utility.setLayer(goAvatar, e_LayerId.OwnPlayer);
}

// 押されたときのコールバック
override protected void onClick(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
        case RENDER_AVATAR:
            //押されたときの処理
            break;
    }
}
// 長押しされたときのコールバック
override protected void onHold(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
        case RENDER_AVATAR:
            //押されたときの処理
            break;
    }
}
```

10.54 CWinCtrlIcon

Script: *ICON*

This is the control to display the icon.

It is possible to display eight texture parts.

This behavior is almost the same as *TEXTURE*. But if you touch it, it becomes darker.

And, unlike *TEXTURE* control, it can have a caption.

10.54.1 Callback reaching the CWindowBase

void onClick (CWinCtrlBase cCtrl)

Parameters **CWinCtrlBase cCtrl** The touched control

void onHold (CWinCtrlBase cCtrl)

Parameters **CWinCtrlBase cCtrl** Control that was pressed and hold

void onDrag (CWinCtrlBase cCtrl, Vector2 pos, Vector2 dragVelocity)

Parameters

- **CWinCtrlBase cCtrl** – The dragged control
- **Vector2 pos** – Drag position, or the position of the mouse cursor
- **Vector2 dragVelocity** – Displacement of the touch position

10.54.2 Example of use

```
// コントロールを取得する方法。
CWinCtrlIcon icnItem = find(ICON_Item) as CWinCtrlIcon;

// 押されたときのコールバック
override protected void onClick(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
        case ICON_Item:
            //押されたときの処理
            break;
    }
}
// 長押しされたときのコールバック
override protected void onHold(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
        case ICON_Item:
            //押されたときの処理
            break;
    }
}
```

10.55 CWinCtrlRecastIcon

Script: *RECASTICON*

It is a control to display a recast icon.

It is possible to display eight texture parts.

This is almost the same behavior as the ICON.

If you make a set of recast value, the specified texture parts do animation which it becomes gradually extending from the bottom. If you set 0 to display as TEXTURE. If you set 1 or more, it hides.

10.55.1 Callback reaching the CWindowBase

`void onClick (CWinCtrlBase cCtrl)`

Parameters `CWinCtrlBase cCtrl` The touched control

`void onHold (CWinCtrlBase cCtrl)`

Parameters `CWinCtrlBase cCtrl` Control that was pressed and hold

`void onDrag (CWinCtrlBase cCtrl, Vector2 pos, Vector2 dragVelocity)`

Parameters

- `CWinCtrlBase cCtrl` – The dragged control
- `Vector2 pos` – Drag position , or the position of the mouse cursor
- `Vector2 dragVelocity` – Displacement of the touch position

10.55.2 Functions / Properties

`CInput::e_State state`

This function returns the state of pressed the icon

`void setRecastTime (int index, float tmRecast)`

Parameters

- `int index` – Texture part index for the recast animation
- `float tmRecast` – Animation time [0-1]

Set the recast time.

`float getRecastTime (int index)`

Parameters `int index` Texture part index for the recast animation

Return float Remaining animation time

Get a recast time.

10.55.3 Example of use

```
// コントロールを取得する方法.
CWinCtrlRecastIcon      rcItem = find(RECASTICON_Item) as CWinCtrlRecastIcon;

// リキャスト表示 (TEX_ID3 のテクスチャを半分表示する)
rcItem.setRecastTime(3, 0.5f);

// 押されたときのコールバック
override protected void onClick(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
        case RECASTICON_Item:
            //押されたときの処理
            break;
    }
}
// 長押しされたときのコールバック
override protected void onHold(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
```

```

    case ICON_ITEM:
        //押されたときの処理
        break;
}
}

```

10.56 CWinCtrlRenderIcon

Script: *RENDERICON*

It is a control for the icon with using Render Texture.

It automatically cut out rectangle from the Render Texture area which you specify.

Multiple rendering result may be included within single Render Texture. After you render to the texture, it can render many icons at low cost.

For example, it is effective to use that when you want to display the dress possible avatar icon on the listbox.

10.56.1 Callback reaching the CWindowBase

void onClick (CWinCtrlBase cCtrl)

Parameters CWinCtrlBase cCtrl The touched control

void onHold (CWinCtrlBase cCtrl)

Parameters CWinCtrlBase cCtrl Control that was pressed and hold

void onDrag (CWinCtrlBase cCtrl, Vector2 pos)

Parameters

- **CWinCtrlBase cCtrl** – The dragged control

- **Vector2 pos** – Drag position

Dragged the control

bool onBeginRenderIcon (CWinCtrlRenderIcon icon)

Parameters CWinCtrlRenderIcon icon The rendered icon control

Return bool Actually choose rendering or not.

It is started rendering on an icon.

Meaning of the return value of this function.

- If you returns true, this control will render to the texture.

- On the other hand, if you return a false, it does not render. And *onEndRenderIcon* is not called.

bool onEndRenderIcon (CWinCtrlRenderIcon icon)

Parameters CWinCtrlRenderIcon icon The rendered icon control

Return bool You select to whether continue the rendering at the next frame.

It is called when the rendering has been completed

Meaning of the return value of this function.

- If you return a true, rendering of the next frame is continued(*onBeginRenderIcon* is called at the next frame.).

- On the other hand, If you return a false, It is not rendered in the next frame.

If you want to resume the stopped rendering , you set the needToRender to true.

10.56.2 Functions / Properties

ulong **regionId**

Please allocate a unique ID other than 0. If this is substituted with 0, reserved rendering area is released.
And if you allocate the same value as the other controls,those controls share the rendering area.

Camera **camera**

This function returns the camera to render texture is assigned. if you want to customize a position or an angle of the camera, please access it.

WinColor **color1**

It is assigned as the background color to render.

The default is (0, 0, 0, 0).

bool **needToRender**

If you want to resume the rendering, please set the true.

onBeginRenderIcon will be called in the next frame.

CInput::e_State **state**

This function returns the state of pressed the icon

10.56.3 Example of use

In this example,you made objects active on *onBeginRenderIcon*.Next,you make them deactivated on *onEndRenderIcon*.

if you do not make these operations,All objects will be rendered to the same icon.

If the *onBeginRenderIcon* returns true, the Camera.Render() is called immediately. If you want to reflect an animation to objects, you must call the Animation.Sample().

When the control is not rendered correctly, try to check the following points.

- Is the layer setting not wrong? / Do you not forget the setting of the layer?
 - カメラにレンダリング対象が収まっているか？
- If objects are not in the camera, they are not rendered.
- Did you set the *regionId* ?

If the *regionId* is 0, it is not performed rendering to render the texture.It is cut out a rectangular area from the render texture in the *regionId* the key, and then rendering.

```
override public bool onBeginRenderIcon(CWinCtrlRenderIcon icon) {
    CWinCtrlListbox lbAvatar = find(CWinCtrlListbox>(LISTBOX_Avatar));
    int idx = lbAvatar.getContentsIndex(icon);
    if (idx < 0) {
        return false;
    }
    GameObject go = m_lstGameObject[idx];
    if (go == null) {
        return false;
    }
    go.SetActive(true);
    Animation anim = go.GetComponent<Animation>();
    anim.Sample();

    return true;
}
override public bool onEndRenderIcon(CWinCtrlRenderIcon icon) {
```

```

CWinCtrlListbox    lbAvatar = lbAvatar = find(CWinCtrlListbox>(LISTBOX_Avatar));
int idx = lbAvatar.getContentsIndex(icon);

GameObject go = m_1stGameObject[idx];
if (go == null) {
    return false;
}
go.SetActive(false);
return false;
}

// 押されたときのコールバック .
override public void onClick(CWinCtrlBase cCtrl) {
switch (cCtrl.id) {
case RENDERICON_Avatar:
    //押されたときの処理.
    break;
}
}

// 長押しされたときのコールバック .
override public void onHold(CWinCtrlBase cCtrl) {
switch (cCtrl.id) {
case RENDERICON_Avatar:
    //押されたときの処理.
    break;
}
}

```

10.56.4 The initial values of each layer

Camera.cullingMask = e_Layer.RenderIcon;

Please do not change it if not particularly necessary.

If the rendering, please be careful to layer settings.

The game object layer should set this value.

```

// gameObjectAvatar が複数の階層でなければ、以下で十分。
gameObjectAvatar.layer = (int) e_LayerId.RenderIcon;
// gameObjectAvatar が複数の階層からできているときは、次の関数を使って再帰的に設定する必要がある。
Utility.setLayer(gameObjectAvatar, e_LayerId.RenderIcon);
// RenderIcon にレンダリングするオブジェクトは、読み込みが終了したタイミングですぐに SetActive(false) しておく必要
gameObjectAvatar.SetActive(false);

```

Note: GameObject.layer set the layer only for a single hierarchy. It does not affect to the children of the hierarchy. To you set the layer in hierarchies of a game objects, you can use the Utility.setLayer. This function can be set recursively layer to root hierarchy and children hierarchies of the game object.

10.56.5 The main flow of rendering

RenderIcon is suitable to display models for each icon. However , the number of layers is limited , it is necessary to reuse a single layer.

This control is rendered by the following procedure in order to solve this problem .

1. Load models(Resources)
2. After you have finished loading , you deactivate the game object by SetActive (false).
3. You set RenderIcon to the layer.

4. You set a unique value to the *regionId*.
5. When OnBeginRenderIcon is called , you make game objects activate.
6. When OnEndRenderIcon is called , you make game objects deactivate.

You can limit game objects to be rendered in RenderIcon. The OnEndRenderIcon returns false. It is good for performance. If a OnEndRenderIcon returns true,it would be to continue rendering.

10.56.6 Timing onBeginRender is called .

- You set a non-zero value to *regionId*.

If you set a unique value other than 0 to *regionId* , *onBeginRenderIcon* is called *onBeginRenderIcon* is continuously called until it returns true. On the other hand , if this function returns false, *onEndRenderIcon* will not be called. You return a false while reading the resource. If it is ready to rendering , you can render resources at the render texture. In this case you return true in this function.(Please do not forget layer settings)

- After the render texture is lost, it becomes necessary to regenerate it.

The reason for render texture is lost is the window minimized,the screen saver,etc.

- After setting the true to *needToRender*
- When the return value of *onEndRenderIcon* is true

If you want to every frame rendering , you can return value of this function to true. However , your application performance is lost.

10.57 CWinCtrlMeter

Script: *METER*

This is the control to display a meter.

You can use this as a progress bar.

It is possible to be displayed eight texture parts.

You can stretch any texture parts.

If the length is set to 1, it display same look as *TEXTURE* .

10.57.1 Callback reaching the CWindowBase

None

10.57.2 Functions / Properties

float meter (int index)

Parameters int index Texture Part Index

Return float Length of the meter

This function returns the percentage of the length from the original size.

It returns a value in the range of 0~1.

It sets 1 by default(Meter state is Max).

void setMeter (int index, float value, float speed = 0.2f)

Parameters

- **int index** – Texture Part Index
- **float value** – Length of the meter(0~1)
- **float speed = 0.2f** – Animation speed

The length of the specified part is changed along with the animation.

10.57.3 Example of use

```
// コントロールを取得する方法。
CWinCtrlMeter mtrLoading = find(METER_Loading) as CWinCtrlMeter;

// メータの位置を取得 (0~1で返ります)
float pos = mtrLoading.meter(3);

// 3番目のテクスチャパートの伸縮を設定 (0~1で指定可能です。)
mtrLoading.setMeter(3, 0.5f);
```

10.58 CWinCtrlScrollbar

Script: *SCROLLBAR*

This is a control to display the scroll bar.

This is displayed by setting the *GROUP* that is a property of the list box.

10.58.1 Callback reaching the CWindowBase

None

10.58.2 Functions / Properties

int index

It specifies the texture index to act as a scroll bar.

Other than the texture specified by this value is displayed without changing the length and position.

float fadeSpeed

This value is the disappearing speed when *display conditions* came off.

10.58.3 Example of use

```
CWinCtrlScrollbar sbListbox = find<CWinCtrlScrollbar>(SCROLLBAR_Listbox);
```

10.59 CWinCtrlListbox

Script: *LISTBOX*

This is the control for the list box.

It has contents of the same number as the number of rows.

It can have a lot of contents as long as the remaining memory.

Not only the list box that is lined with content in the vertical direction , it is possible to arrange also in the horizontal direction .

10.59.1 Callback reaching the CWindowBase

`void onDrag (CWinCtrlBase cCtrl, Vector2 pos, Vector2 dragVelocity)`

Parameters

- `CWinCtrlBase cCtrl` – The dragged control
- `Vector2 pos` – Drag position , or the position of the mouse cursor
- `Vector2 dragVelocity` – Displacement of the touch position

Called when you scroll . Even if you lock the scroll , this callback is called .

10.59.2 Functions / Properties

`Vector2 viewsize`

the size of list box. This is the same as the size property.

`Vector2 screensize`

It is a virtual screen size of the list box. It becomes the sum total of all of the height of the contents list.

`Vector2 offset`

It is a place that is a point of virtual screen. Move at once If you set the value. If you'd like to scroll it smoothly , you can use `setSmoothOffset` .

`bool isSwipe`

Get the state whether is swiped.

`void setSmoothOffset (Vector2 offset, float spd)`

Parameters

- `Vector2 offset` – offset
- `float spd` – Speed

`Vector2 smoothOffset`

Same behavior as `setSmoothOffset` with 0.25 speed

`bool isSmoothScrolled`

Get the state whether is scrolled by setSmoothOffset.

`Vector2 getContentsOffset (int index, e_Anchor eAnchor)`

Parameters

- `int index` – The index of contents offset
- `e_Anchor eAnchor` – The position of contents is specified with an anchor.

You can get the offset which is displayed along the anchor.

If the contents of the list box are arranged in a vertical direction

<code>e_Anchor</code>	<code>Location</code>
Bottom,LeftBottom,RightBottom	Offset value for the contents to be aligned with the bottom of the view.
Top,LeftTop,RightTop	Offset value for the contents to be aligned with the top of the view
Center,Left,Right	Offset value for the contents to be aligned with the center of the view

If the contents of the list box are arranged in a horizontal direction

e_Anchor	Location
Left,LeftBottom,LeftTop	Offset value for the contents to be aligned with the left of the view
Right,RightBottom,RightTop	Offset value for the contents to be aligned with the right of the view
Center,Top,Bottom	Offset value for the contents to be aligned with the center of the view

10.59.3 Example of use

```
override public void onUpdate() {
    // コントロールを取得する方法.
    CWinCtrlListbox lbFriends = find<CWinCtrlListbox>(LISTBOX_Friends);

    // リストボックスの行数を設定.
    lbFriends.resize(10);

    // リストボックス内の状態を更新する (この場合 10 行更新することになる)
    for (int i = 0; i < lbFriends.Count; ++i) {
        // リストボックスのコンテンツを一行取得する.
        CWinContents contents = lbFriends.getContentsFromIndex(i);

        //BUTTON(Name) を取得
        CWinCtrlButton btnName = contents.find<CWinCtrlButton>(BUTTON_Name);

        //ICON(Icon) を取得
        CWinCtrlIcon icnName = contents.find<CWinCtrlIcon>(ICON_Icon);
    }
}
```

10.60 CWinCtrlListboxEx

Script: *LISTBOXEX*

This is the control for the list box.

It has contents of the same number as the number of rows.

It can have a lot of contents as long as the remaining memory.

Not only the list box that is lined with content in the vertical direction , it is possible to arrange also in the horizontal direction .

Differences between the *LISTBOX* is to measure the content size automatically.

If the content is arranged in a vertical direction, it automatically calculate only height.

If the content is arranged in a horizontal direction, it automatically calculate only width.

10.60.1 Callback reaching the CWindowBase

void **onDrag** (CWinCtrlBase *cCtrl*, Vector2 *pos*, Vector2 *dragVelocity*)

Parameters

- **CWinCtrlBase cCtrl** – The dragged control
- **Vector2 pos** – Drag position , or the position of the mouse cursor
- **Vector2 dragVelocity** – Displacement of the touch position

Called when you scroll . Even if you *lock* the scroll , this callback is called .

10.60.2 Functions / Properties

Vector2 **viewsize**

the size of list box. This is the same as the size property.

Vector2 **screensize**

It is a virtual screen size of the list box. It becomes the sum total of all of the height of the contents list.

Vector2 **offset**

It is a place that is a point of virtual screen. Move at once If you set the value. If you'd like to scroll it smoothly , you can use *setSmoothOffset* .

bool **isSwipe**

Get the state whether is swiped.

void setSmoothOffset (Vector2 offset, float spd)

Parameters

- **Vector2 offset** – offset
- **float spd** – Speed

Vector2 **smoothOffset**

Same behavior as *setSmoothOffset* with 0.25 speed

bool **isSmoothScrolled**

Get the state whether is scrolled by *setSmoothOffset*.

Vector2 getContentsOffset (int index, e_Anchor eAnchor)

Parameters

- **int index** – The index of contents offset
- **e_Anchor eAnchor** – The position of contents is specified with an anchor.

You can get the offset which is displayed along the anchor.

If the contents of the list box are arranged in a vertical direction

e_Anchor	Location
Bottom,LeftBottom,RightBottom	Offset value for the contents to be aligned with the bottom of the view.
Top,LeftTop,RightTop	Offset value for the contents to be aligned with the top of the view
Center,Left,Right	Offset value for the contents to be aligned with the center of the view

If the contents of the list box are arranged in a horizontal direction

e_Anchor	Location
Left,LeftBottom,LeftTop	Offset value for the contents to be aligned with the left of the view
Right,RightBottom,RightTop	Offset value for the contents to be aligned with the right of the view
Center,Top,Bottom	Offset value for the contents to be aligned with the center of the view

10.60.3 Example of use

```
override public void onUpdate() {
    // コントロールを取得する方法.
    CWinCtrlListboxEx    lbPartys = find<CWinCtrlListboxEx>(LISTBOXEX_Partys);

    // リストボックスの行数を設定.
    lbPartys.resize(10);

    // リストボックス内の状態を更新する (この場合 10 行更新することになる)
    for (int i = 0; i < lbPartys.Count; ++i) {
        // リストボックスのコンテンツを一行取得する.
        CWinContents contents = lbPartys.getContentsFromIndex(i);
```

```

//BUTTON(Name) を取得
CWinCtrlButton btnName = contents.find<CWinCtrlButton>(BUTTON_Name);

//ICON(Icon) を取得
CWinCtrlIcon icnName = contents.find<CWinCtrlIcon>(ICON_Icon);

//LISTBOX(Party) を取得
CWinCtrlListbox lbParty = contents.find<CWinCtrlListbox>(LISTBOX_Party);
}
}

```

10.61 CWinCtrlContainer

Script: *CONTAINER*

It is a control for grouping multiple controls .

Size of the container is determined by the SIZE. The size of the virtual screen is determined by CONTENTS_SIZE.

When virtual screen size is larger than the display size , it is possible to scroll the controls included area.

10.61.1 Callback reaching the CWindowBase

void **onDrag** (CWinCtrlBase *cCtrl*, Vector2 *pos*, Vector2 *dragVelocity*)

Parameters

- **CWinCtrlBase cCtrl** – The dragged control
- **Vector2 pos** – Drag position , or the position of the mouse cursor
- **Vector2 dragVelocity** – Displacement of the touch position

When you scrolled the area,this function is called.

10.61.2 Example of use

```

// コントロールを取得する方法.
CWinCtrlContainer      ctnMap = find<CWinCtrlContainer>(CONTAINER_Map);
// コンテンツ内のコントロールを取得する方法
// コンテナ内のコンテンツも次の通り、findでアクセス可能です
CWinCtrlTexture txMap = find<CWinCtrlTexture>(TEXTURE_Map);

```

10.62 CWinCtrlFrame

Script: *FRAME*

This is the control to display the frame.

Unlike bar, it does not have a caption.

10.62.1 Callback reaching the CWindowBase

void **onClick** (CWinCtrlBase *cCtrl*)

Parameters CWinCtrlBase *cCtrl* The touched control

void **onHold** (CWinCtrlBase *cCtrl*)

Parameters CWinCtrlBase *cCtrl* Control that was pressed and hold

void **onDrag** (CWinCtrlBase *cCtrl*, Vector2 *pos*, Vector2 *dragVelocity*)

Parameters

- CWinCtrlBase *cCtrl* – The dragged control
- Vector2 *pos* – Drag position , or the position of the mouse cursor
- Vector2 *dragVelocity* – Displacement of the touch position

10.62.2 Example of use

```
// コントロールを取得する方法.
CWinCtrlFrame ctrlStart = find(FRAME_BG) as CWinCtrlFrame;

// 押されたときのコールバック
override protected void onClick(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
        case FRAME_BG:
            //押されたときの処理
            break;
    }
}
```

10.63 CWinCtrlLabel

Script: *LABEL*

This is the control to display the label.

10.63.1 Callback reaching the CWindowBase

None

10.63.2 Example of use

```
// コントロールを取得する方法.
CWinCtrlLabel ctrlStart = find(LABEL_Title) as CWinCtrlLabel;
```

10.64 CWinCtrlBar

Script: *BAR*

The control to display a bar .

It can have a caption.

10.64.1 Callback reaching the CWindowBase

void **onClick** (CWinCtrlBase *cCtrl*)

Parameters CWinCtrlBase *cCtrl* The touched control

void **onHold** (CWinCtrlBase *cCtrl*)

Parameters CWinCtrlBase *cCtrl* Control that was pressed and hold

10.64.2 Example of use

```
// コントロールを取得する方法.
CWinCtrlBar ctrlStart = find<CWinCtrlBar>(BAR_Title);

// 押されたときのコールバック
override public void onClick(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
        case BAR_Title:
            //押されたときの処理
            break;
    }
}
```

How to change the default value:

10.65 How to change the default value

“KsSoft/Plugins/KsSoftConfig.cs” is set various default values.

When you want to change the default value, please fix this file.

10.65.1 Asset bundle

string **AssetbundlePath**

Change the output folder of the asset bundle.

string **AssetbundleExt**

Extension of the asset bundle file.

string **ResourcePath**

Folder path to be output as a resource data.

string **deugPath**

It is used when loading the asset bundle via “file://…”.

Note: Invalid value in the application.

10.65.2 Window resource

uint **WindowAssetbundleId**

Window asset bundle ID to download by default

uint **DefaultFontKind**

The default value for the font ID is used when not specified.

uint **DefaultTextureId**

The default value for the texture ID is used when not specified.

10.65.3 Fonts

```
uint FontsAssetbundleId
      Asset bundle ID, which is using a font group when asset bundling
```

10.65.4 Message data sheet

Dictionary<e_Locale, uint> **MessageDataId**

When you add a locale data, please add the information below. Value is the asset bundle ID to be used when asset bundling the locale data.

```
static private Dictionary<e_Locale, uint> m_dicStrMessageDataMulId = new Dictionary<e_Locale, uint> {
    {e_Locale.JP, MulId.Id (0,2,0)},
    {e_Locale.EN, MulId.Id (0,2,1)},
};
```

10.65.5 Address/IP

string **httpserver** = “http://xxx.xxx.xxx.xxx/”

Set the address when you download the asset bundle via HTTP

Indices and tables

- genindex
- modindex
- search