WindowSystem Documentation

Release 2.00

Seikichi Kanemura,KsSoft

May 17, 2019

CONTENTS

CHAPTER

ONE

FEATURE

1.1 Features of window system

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

> You can now specify a percentage of the parent's size for most positions, offsets, and sizes.

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.

CHAPTER

TWO

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..

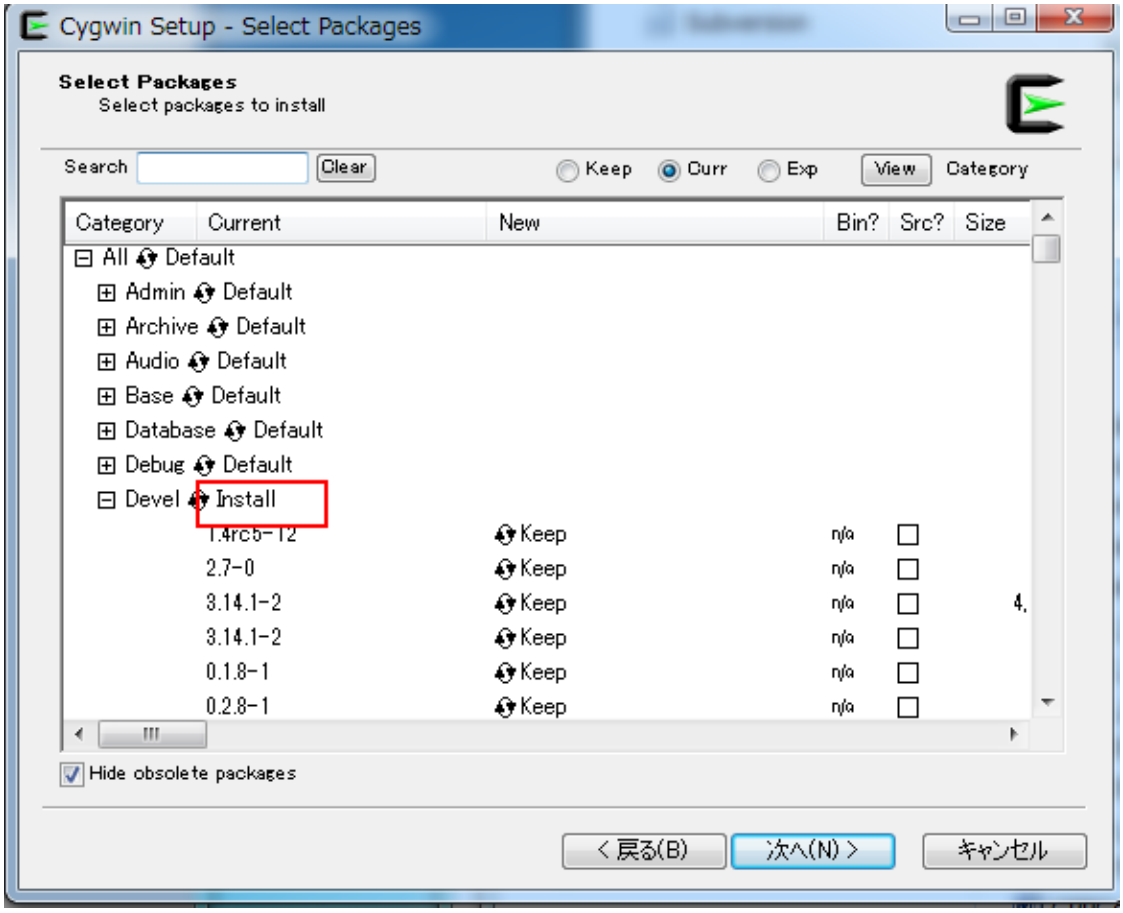
Download the cygwin

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 .

Add the system path to “bin” of the installed folder .

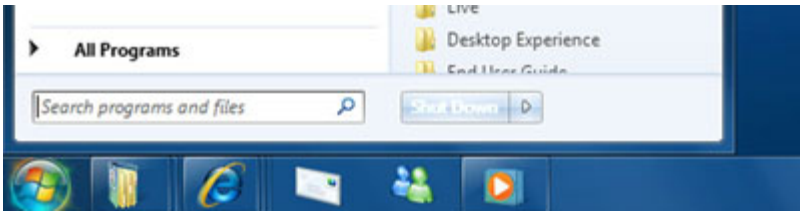
It has become in the following folder by the default

```
C:\cygwin\bin or  c:\cygwin64\bin(64bit version)
```

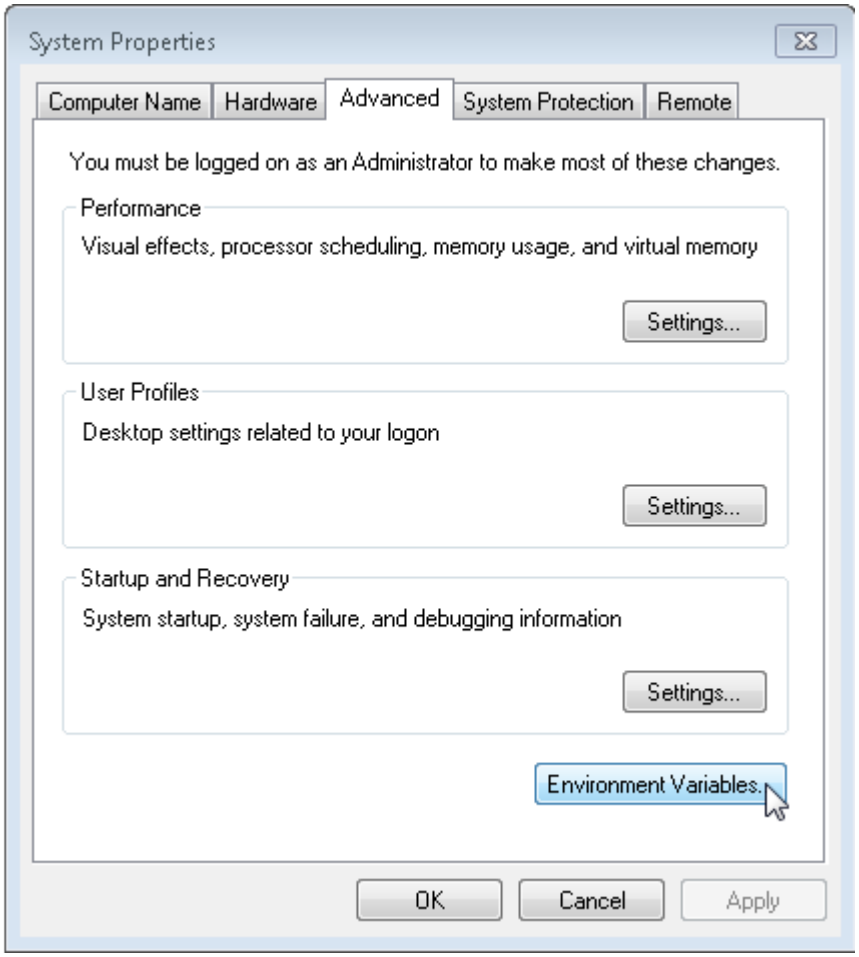
Note: It is not the environment variable of cygwin, it is the system environment variable of windows.

Detailed procedure is as follows.

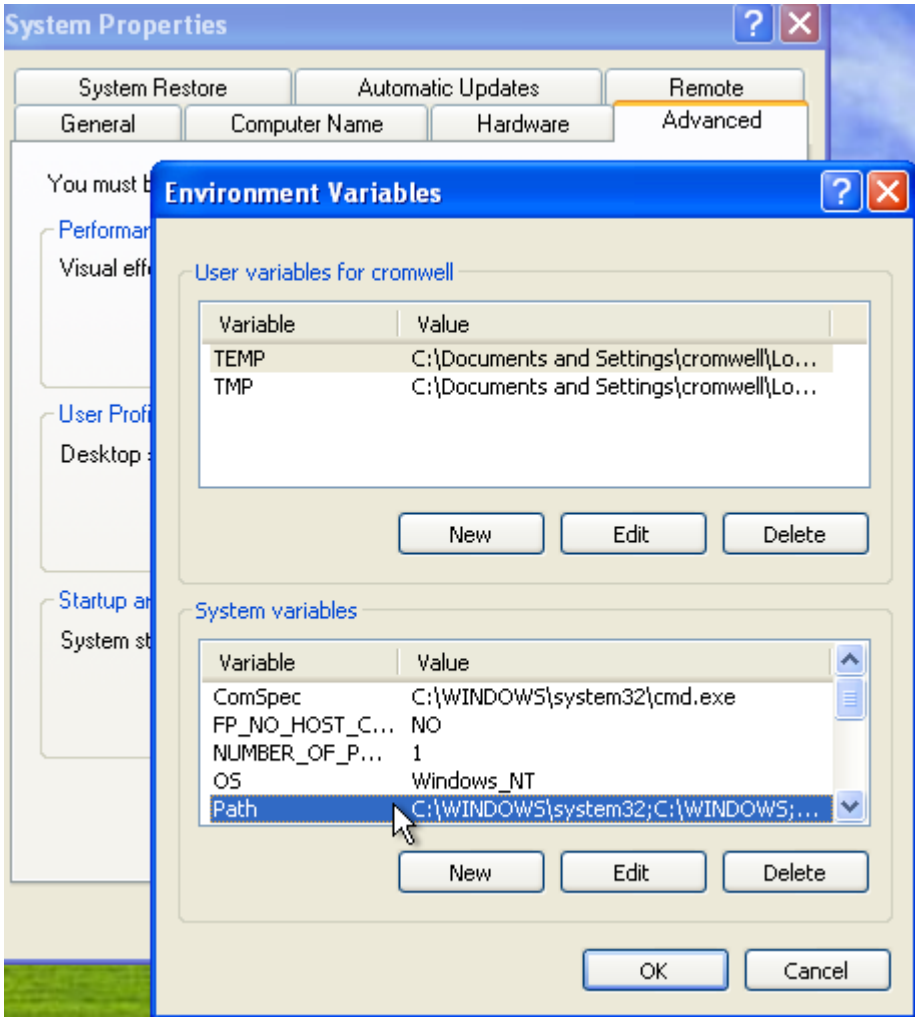
Click the start menu of windows.



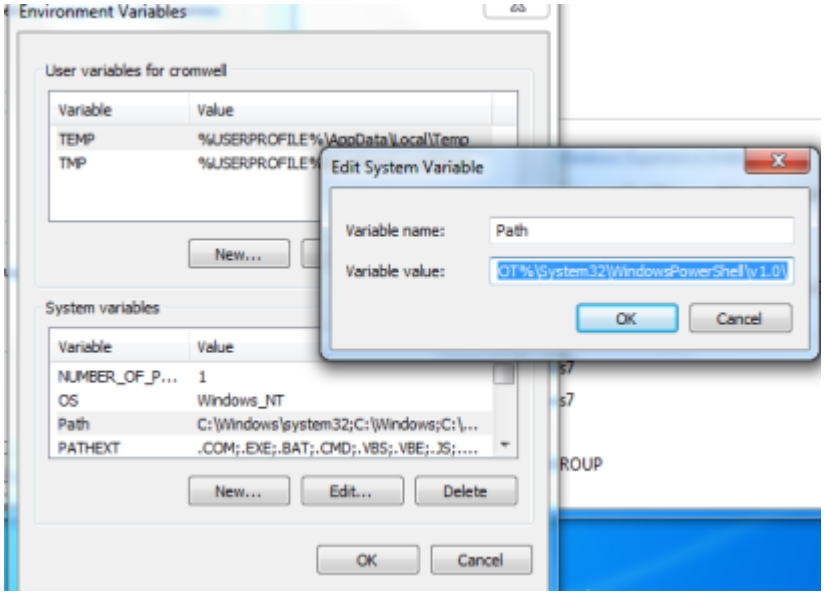
Input “sysdm.cpl” in Search programs and files. New window that named System Properties will appear.



Select the Advanced panel, and click the Enviroment Variables button.



Now that you are looking at the list of system environment variables, find the Path value in the list. You may have to scroll down to find it. Click on the Path value to highlight it as shown here, and then click the Edit button.



A new window will appear with the current value highlighted. Do not remove or change the current value, or you will wreck your user settings!

You will see that the PATH variable already has a default value of something like the following:

```
%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem
```

The semicolons separate the values, so there are three components in this default PATH:

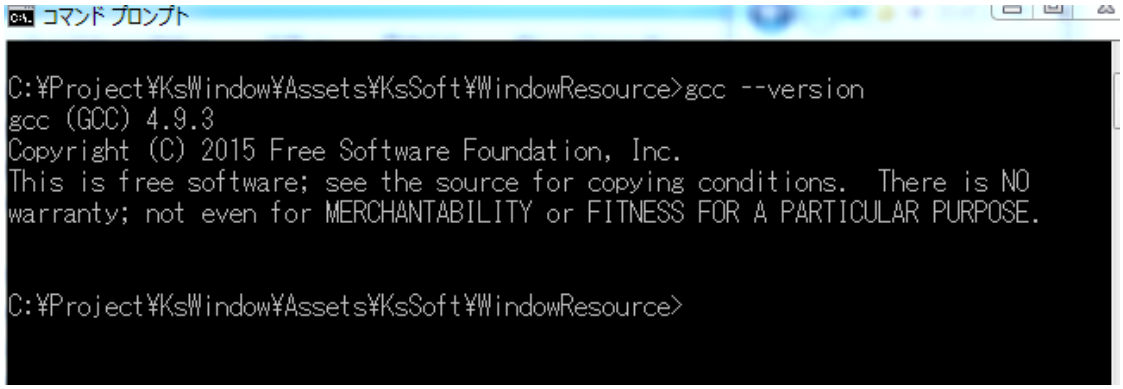
```
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
```

You want to add ;c:\cygwin\bin to the end of that list, so it becomes something like this. However, do not change the entire string to this example! Leave the existing PATH assignment as you find it and simply add your new directories.

```
%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;c:\cygwin\bin
```

The confirmation

Open a windows command prompt(not the cygwin's bash!), try typing "gcc –version". If you have been installed,it is displayed as follows.



When you can not get the same result, you try to check two things.

1. You confirm to exist gcc.exe in the "C:\cygwin\bin" directory. If you can not find gcc.exe, please rechecked *here.* .
2. If you find gcc.exe at the "C:\cygwin\bin" directory, please rechecked *here* .

2.1.2 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.

2.1.3 A preparation for the multi-lingual

Install the Python 2.7

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.

Install the Python2.7

Add the windows system path for the Python.

Add the system path of the installed folder .

Please refer to here how to set the python path.

The path depends on the version of Python you have installed.The default path is:

```
C:\Python3x
```

Install the Python 2.7

This multiligual message data tool used the **openpyxl** library. you need to install this library.Install the openpyxl on the command prompt as follows.

```
pip install pywin32
```

2.2 window system's samples

2.2.1 Sample scenes

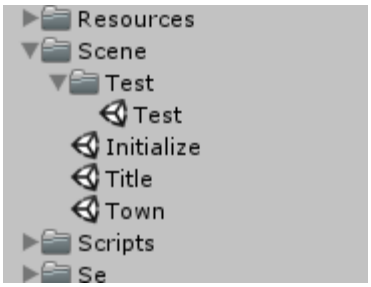
I prepare two samples.

- The sample which you can confirm each control behaviour.

  KsSoft/Scene/Test/Test.unity

- The sample which assumed that you used this in your application.

  Run "KsSoft/Scene/Initialize.unity".



The sample which you can confirm each control behaviour.

This is the sample that you can confirm behavior of almost all control.

On the Console, this sample outputs logs in console when various callbacks were sent. You can see when a window receive various callback. And dragable controls are able to drag by the description of wra scripts.

Related codes are as follows.

- KsSoft/Script/Test/Test.cs
- KsSoft/Script/Test/CWinTest.cs
- KsSoft/Script/Test/CWinTestBase.cs
- KsSoft/WindowResource/CWinTest.wra

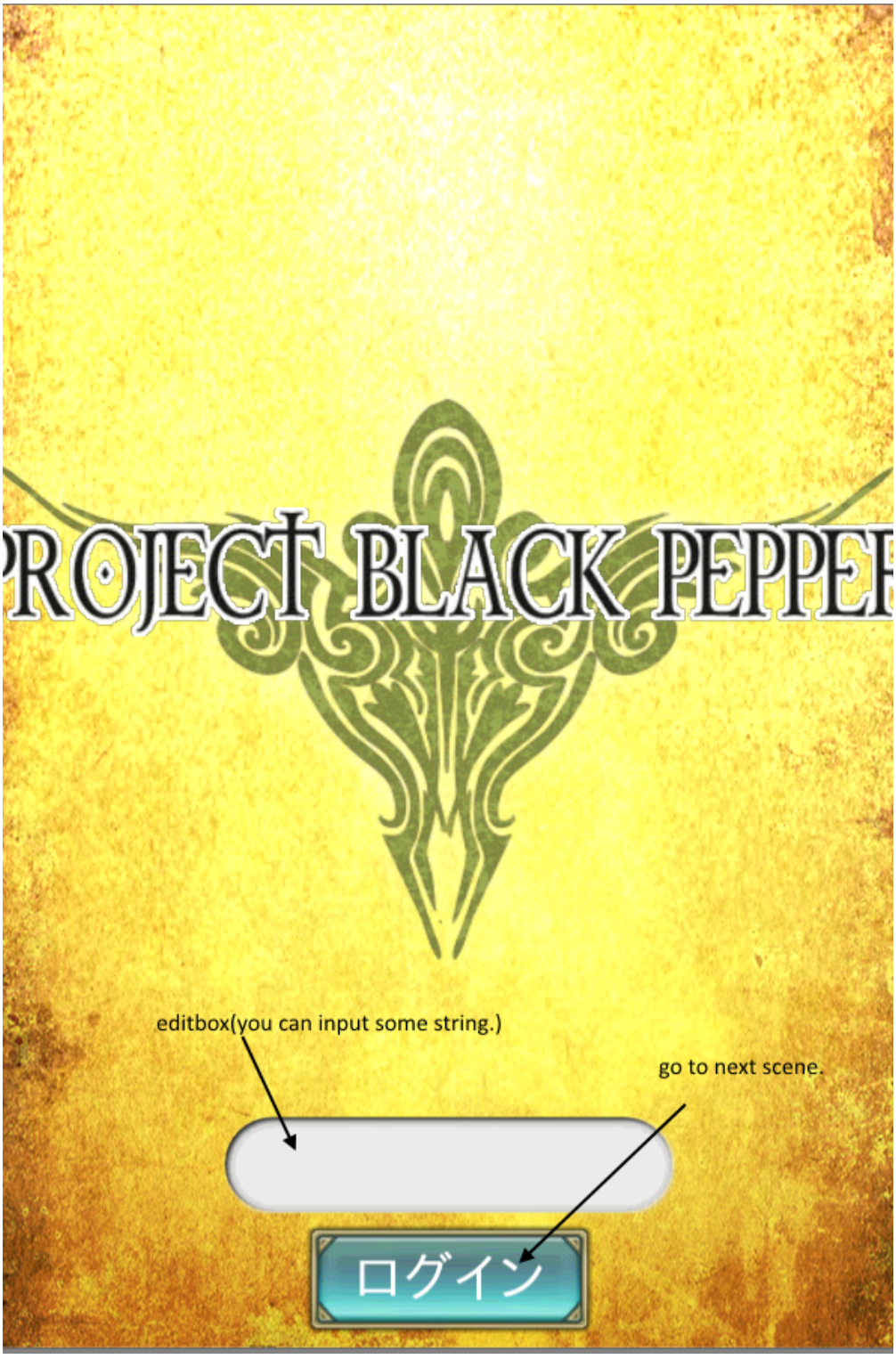
Note: After you edit the CWinTest.wra, if you want to check changes, you need to do *Export Window Resource* .

The sample which assumed that you used this in your application.

2.2.2 The title

CWinTitle.cs

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



2.2.3 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

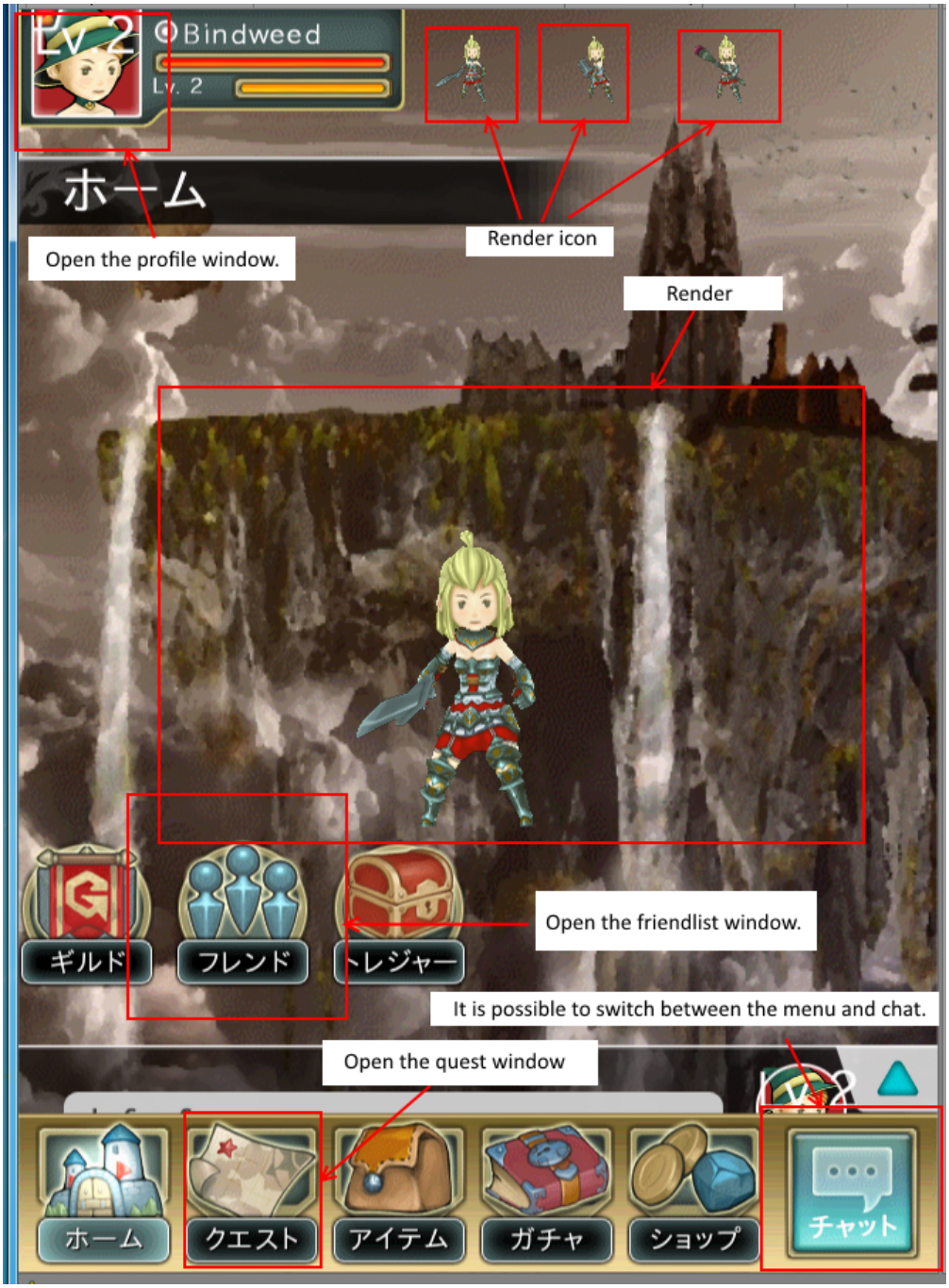
- KsSoft/Script/Scene/Town/CWinBottomPart.cs
- KsSoft/Script/Scene/Town/CWinBottomPartBase.cs
- KsSoft/WindowResource/CWinBottomPart.wra

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.



2.2.4 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.

2.2.5 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* .

2.2.6 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.

2.2.7 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.

2.2.8 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 “Drop the asset bundle from the server” vs “Using streaming assets”

Use streaming assets (Default)

To stream an asset bundle:

Assets/KsSoft/Plugins/KsSoftConfig.cs, line = 27

```
public class KsSoftConfig : KsSoftConfigBase {
    static public void initialize() {
      m_UseStreaming = true;
    }
```

this is default value.

Download the asset bundle from the server.

If your application has to download data via HTTP using asset bundle, please set it as follows.

Assets/KsSoft/Plugins/KsSoftConfig.cs, line = 27

```
public class KsSoftConfig : KsSoftConfigBase {
    static public void initialize() {
      m_UseStreaming = false;
    }
```

2.3.2 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* .

2.3.3 Output all of the asset bundle

From the [menu] [Tools]->[KsSoft]->[Export All]->[Windows,Mac]

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

Note: When it is configured to *use the resources* , the window system exports the resource data rather than assetbundles .

2.3.4 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.

wra files/wrb files

wra is an abbreviation of Window Resource Ascii. This script file define Control and View for a window.

wrb is an abbreviation of Window Resource Binary.If you succeed to compile a wra file on Unity Editor, you can get the wrb file.

The include path

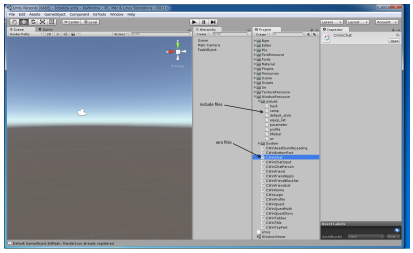
The include path is set to the following path.

```
Assets/KsSoft/WindowResource/include
```

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

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 “Assets/KsSoft/WindowResource/wrb”. These files are the result of the compiled your script files.

2.3.5 Window viewer

You can just preview the compiled windows in the viewer.

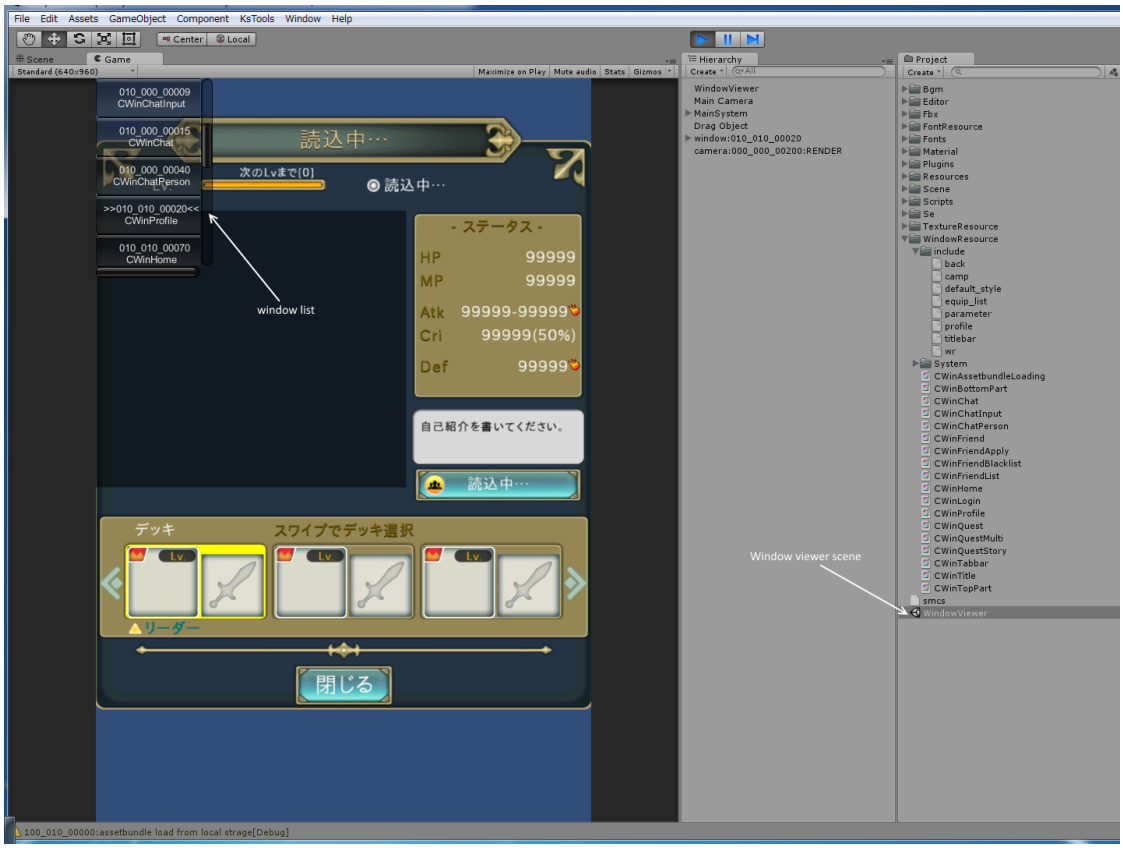
In addition,you can preview changes that the window was successful to compile once without stopping the viewer.

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.

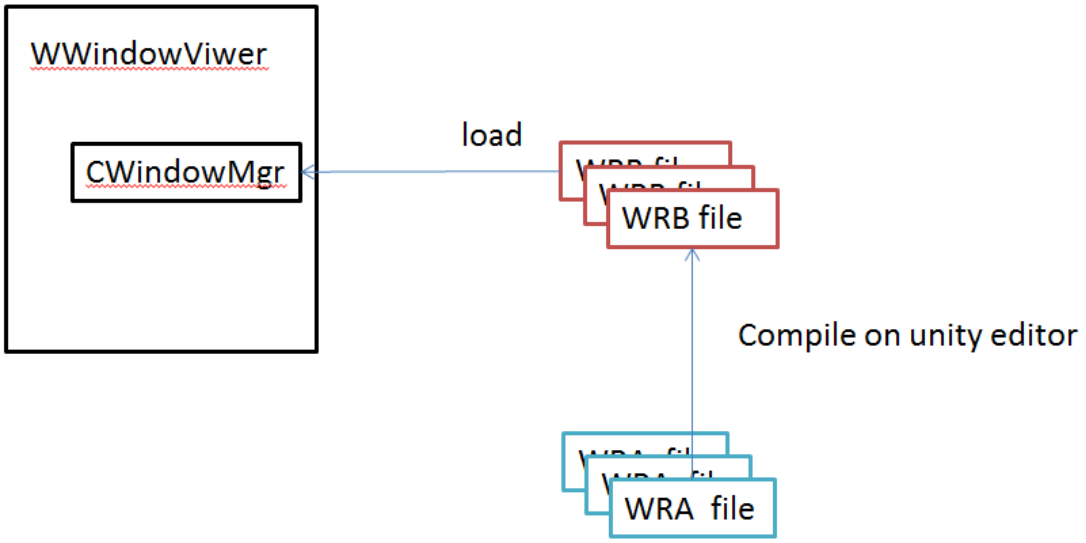
   You can compile a your window script in the viewer scene running.

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



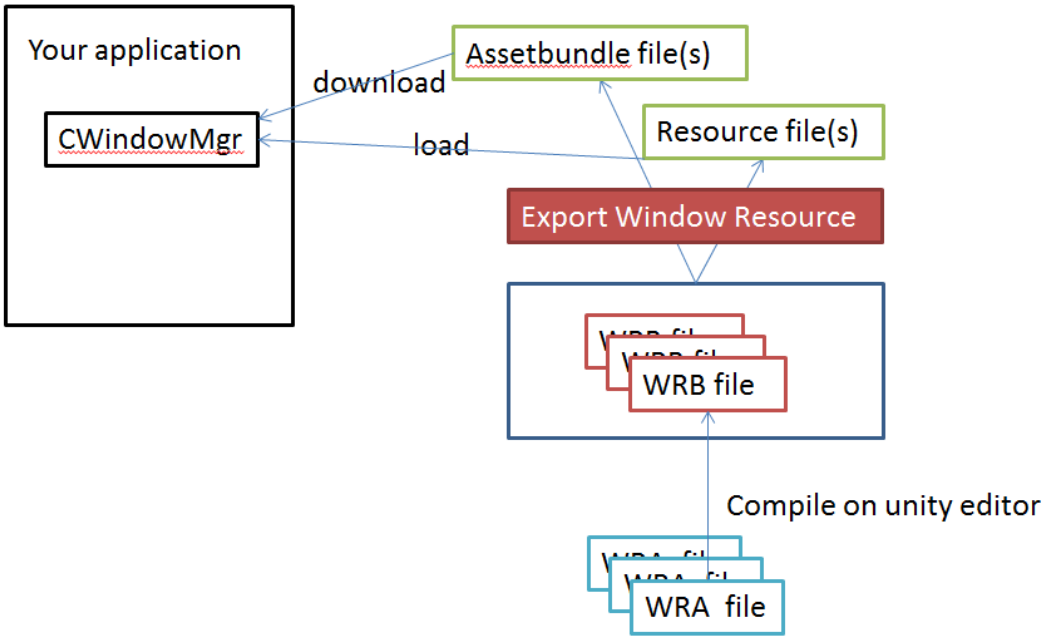
2.3.6 Assetbundles/Resources

The WWindowViewer directly loads wrb files which were compiled from wra files. You can check changes of wra file as soon after wra file was compiled.



But it is burdensome to put or load many wrb files on your application. This window system can bundle compiled wra files, and it can export resources(or a resource) and assetbundle files(or an assetbundle file).If your application preliminarily load the bundled file(s), you can create all windows at any time.

This bundled file can be loaded via HTTP or Resource.



Select the [Tools]->[KsSoft]->[Export Window Resource] menu on the Unity. An assetfile is output to the "assetbundles" folder.

The assebundle file name is **000_000_00010.unity3d** as default.

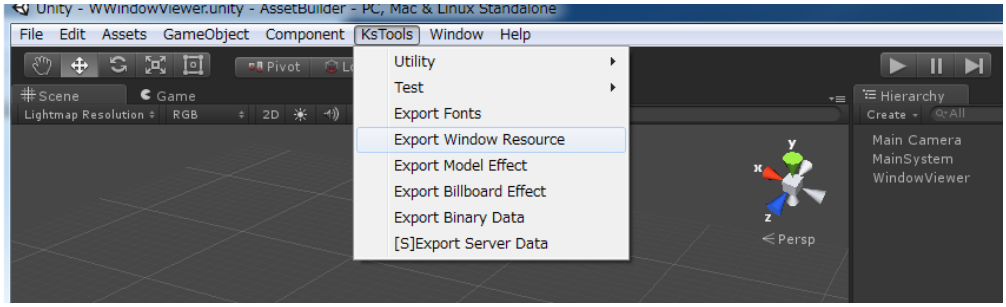
If you want to change the file to load, please refer to *here* .

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.

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

WINDOW property: *RESOURCE = path* 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.



2.3.7 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>();
    }
    //=========================================================================
    /*!Initialize.
```

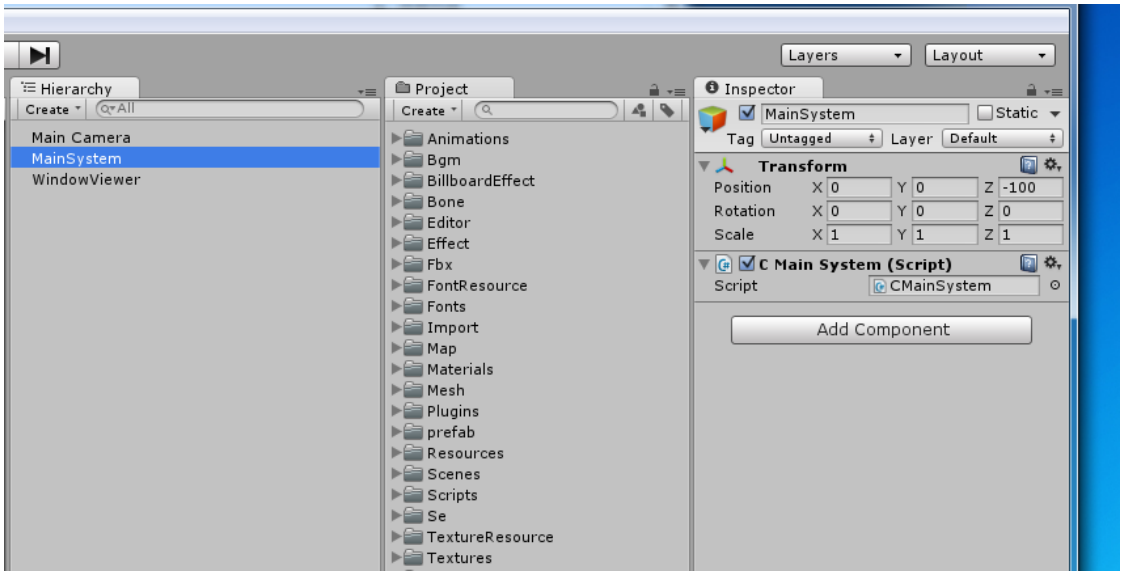
(continues on next page)

(continued from previous page)

```
     * @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 = NOTITLEBAR|NOFRAME|ANCHOR_CENTER|TOP;
   SIZE = {100},{100};
   PRIORITY = PROGRESSBAR_PRIORITY;
};

METER(ProgressTotal) {
   ID = 000_001_00000;
   STYLE = ANCHOR_BOTTOM;
   POSITION = 0,$y;
   SIZE = -128{100};
   TEX_ID = 0,"MTRB";
   COLOR = COLOR32(255.0,255.0,255.0,255.0);
   SIZE1 = -128{100};
   TEX_ID1 = 0,"MTR";
   COLOR1 = COLOR32(255.0,255.0,255.0,255.0);
};
$y -=64;
METER(ProgressPart) {
   ID = 000_001_00010;
   STYLE = ANCHOR_BOTTOM;
   SIZE = -128{100},-32;
   POSITION = 0,$y;
   TEX_ID = 0,"MTRB";
   COLOR = COLOR32(255.0,255.0,255.0,255.0);
   SIZE1 = -128{100},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;
```

(continues on next page)

(continued from previous page)

```
   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.windowId);
if (cLoading != null) {
   //find created this window.
} else {
   //can't find this window.
}
```

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 CSeResourceMgr .

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

CSeResource 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>();
        if (KsSoftConfig.UseAssetBundle) {
                gameObject.AddComponent<CAssetBundleMgr>();
            }
        gameObject.AddComponent<CSpriteFontMgr>();
        gameObject.AddComponent<CTextureResourceMgr>();
        gameObject.AddComponent<CWindowMgr>();
        gameObject.AddComponent<CBgmResourceMgr>();
        gameObject.AddComponent<CSeResourceMgr>();
        gameObject.AddComponent<CSoundEffectMgr>();
    }
    //=========================================================================
    /*!Initialize.
    * @brief   initialize
    */
    override protected void initialize() {
        base.initialize();
        //-----------------------------------------
        // WindowMgr initilaize.
        //-----------------------------------------
        CWindowMgr     cWindowMgr = CWindowMgr.Instance;
        // Assign standard SEs.
        cWindowMgr.soundeffect = CSeResourceMgr.Instance.reference(new MulId(52,0,0),
↪true);    //2D sound;
        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;
            }
```

(continues on next page)

(continued from previous page)

```
        }
    }
}
```

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 **:class:'CMessageDataSheetMgr<CMessageDataSheetMgr>'** .

Since **:class:'CMessageDataSheet<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<CInput>();
        if (KsSoftConfig.UseAssetBundle) {
                gameObject.AddComponent<CAssetBundleMgr>();
            }
        gameObject.AddComponent<CSpriteFontMgr>();
        gameObject.AddComponent<CTextureResourceMgr>();
        gameObject.AddComponent<CWindowMgr>();
        gameObject.AddComponent<CBgmResourceMgr>();
        gameObject.AddComponent<CSeResourceMgr>();
        gameObject.AddComponent<CSoundEffectMgr>();

        addManager(new CMessageDataSheetMgr(Utility.getSystemLocale()));
    }
    //=============================================================================
    /*!Initialize
    * @brief   initialize
    */
    override protected void initialize() {
        base.initialize();
```

(continues on next page)

(continued from previous page)

```
        //-----------------------------------------
        // WindowMgr initialize.
        //-----------------------------------------
        CWindowMgr     cWindowMgr = CWindowMgr.Instance;
        cWindowMgr.captiondata = CMessageDataSheetMgr.Instance.find(new FiveCC("WNDW
↪"));
        // Assign standard SEs
        cWindowMgr.soundeffect =  CSeResourceMgr.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.

CHAPTER

THREE

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 uppper,uint middle,uint lower);
MulId(uint val);

// Properties
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 thease are encoded in the same numerical value.

```
"BTN00"
"BTN0?" = "BTN0_"
"Test"
```

Access method from C#

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

```
//Constructor
FiveCC(string sId);
FiveCC(uint uppper,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);
```

CHAPTER

FOUR

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

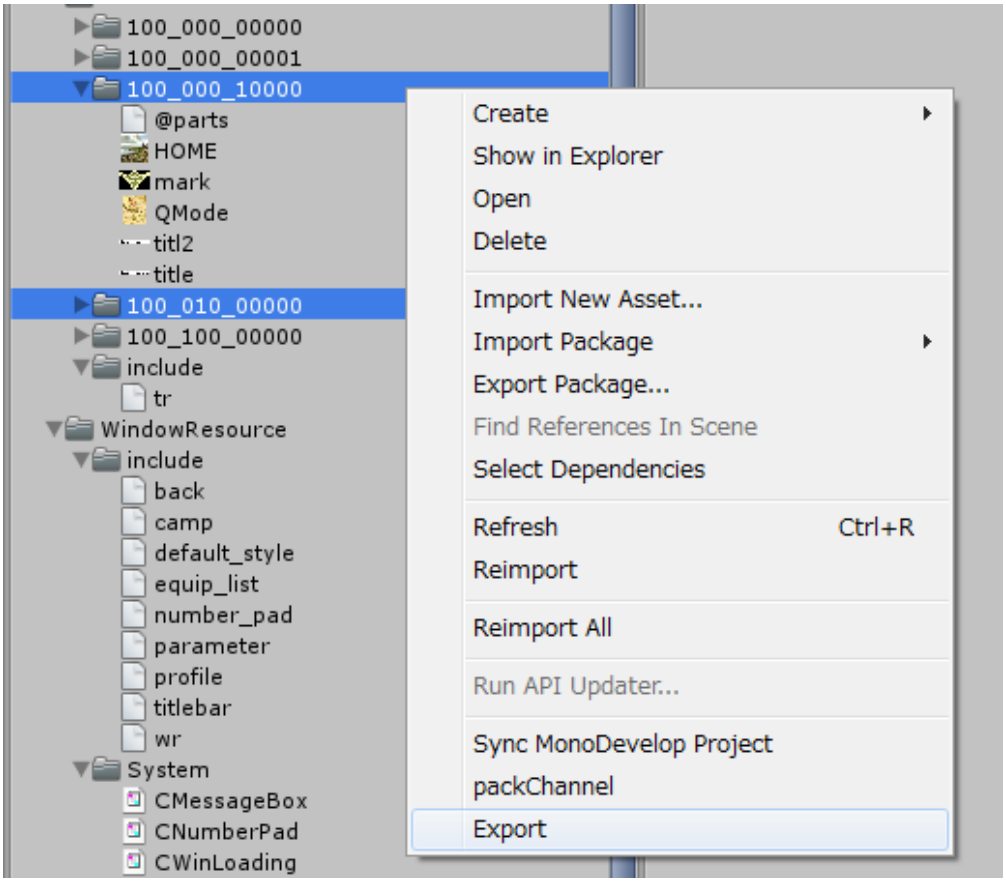
It can also be called a "part". This is part of the texture atlas.

The texture

When referred to as a texture in this document ,it is the summraized 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.pgn
000_000_00010.png
```

The folder name constraints

It must be *MulID* .

Asset bundles because thease 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 create two files as follows under the "KsSoft/Resources/" folder.

- 001_000_00000.spr
- 001_000_00000.tex

This is not loaded by assetbundle,It is loaded this two file from resources. The default is that output to assetbundles.

```
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("blck") {
    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;
};
```

(continues on next page)

(continued from previous page)

```
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.

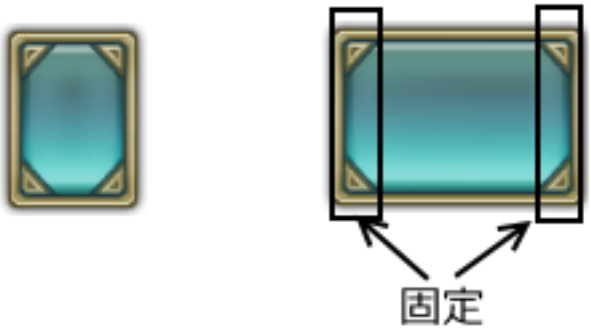
Division of parts default has become NODIVIDE.

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;
};
```

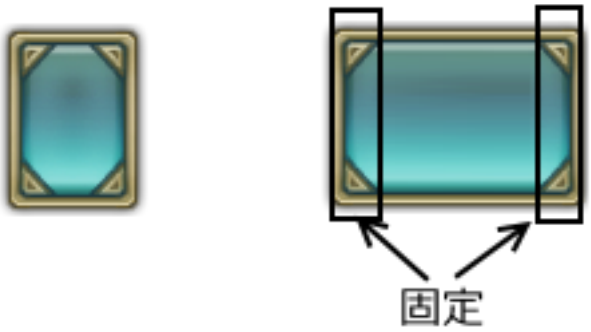


DIVIDE3V = Top fixed height,Bottom fixed heighth

PATCH3V = Top fixed height,Bottom fixed heighth

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;
};
```

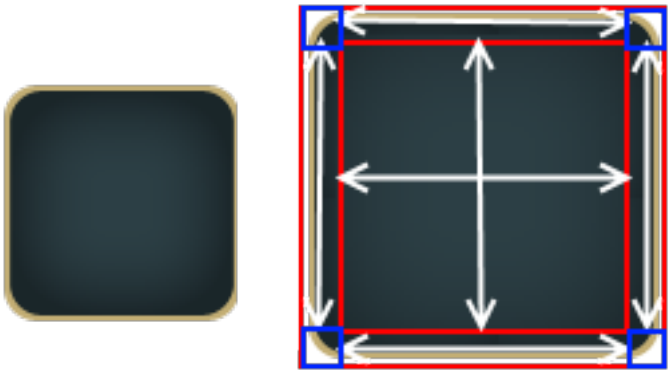


DIVIDE9 = Top left fixed width, Top left fixed height, Bottom right fixed width, Bottom right fixed high

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

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
```

CHAPTER

FIVE

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 A channel of the bitmap file. Four A 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.

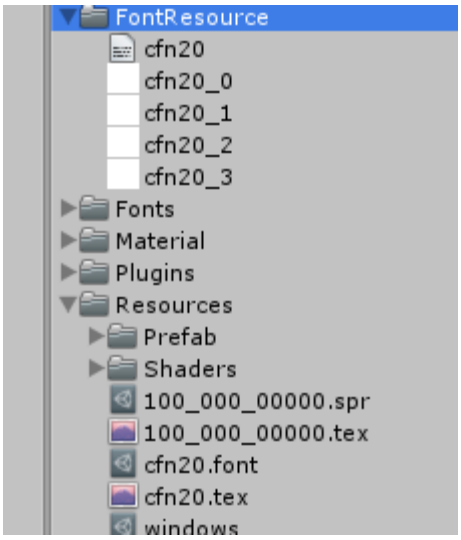
If you want to create a font named "cfn20", you need to prepare the files such as follows.And you place these in this location.

- 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

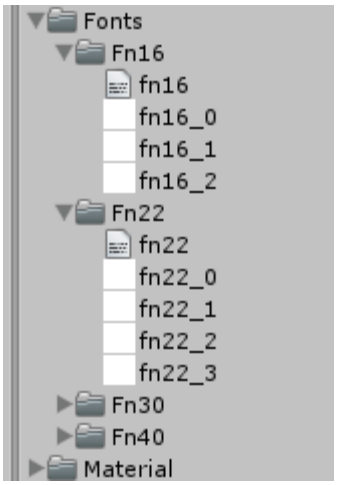
Also, when you want to use multiple fonts, put the number required by the same procedure.

If you are ready, run the [Tools]->[KsSoft]->[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

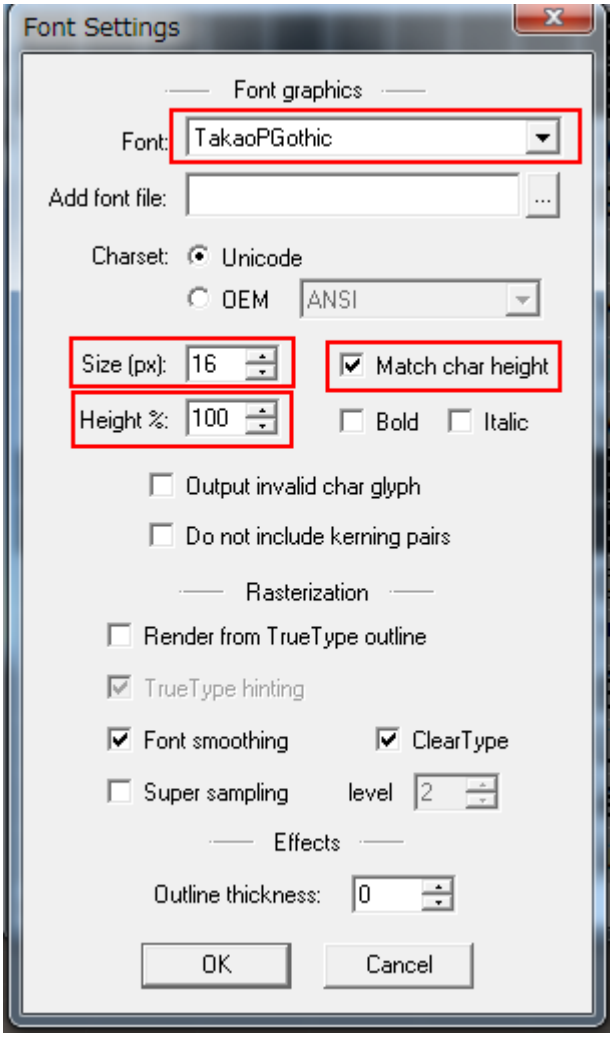
Font Settings

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].

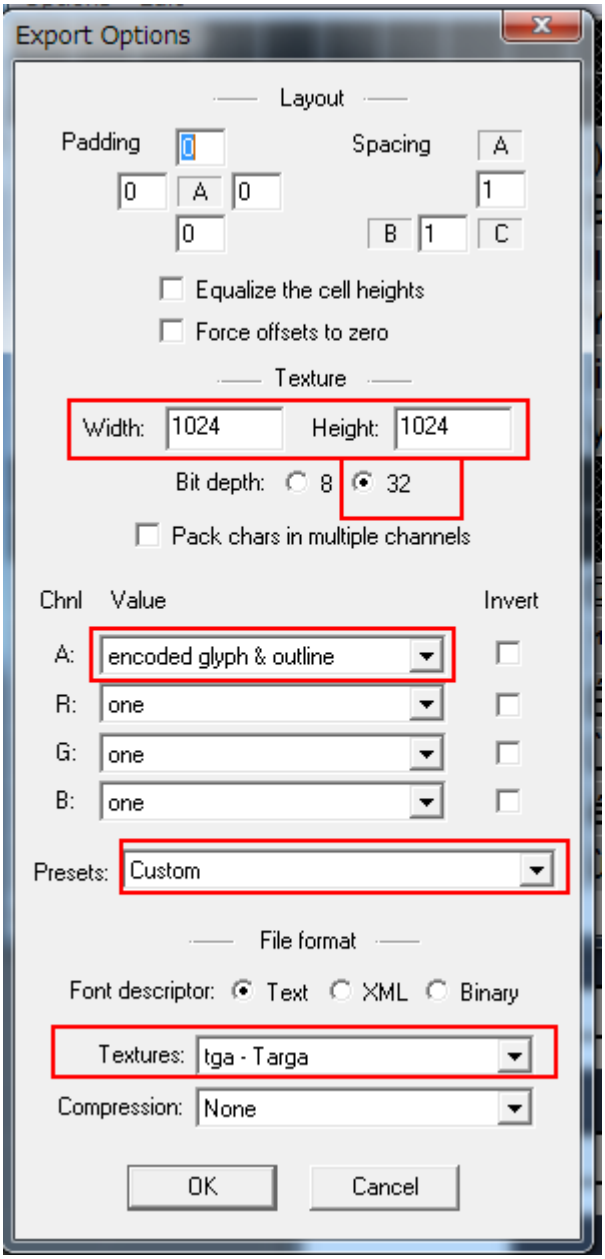
Put 100 to [Hieght%].



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”.

CHAPTER

SIX

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.

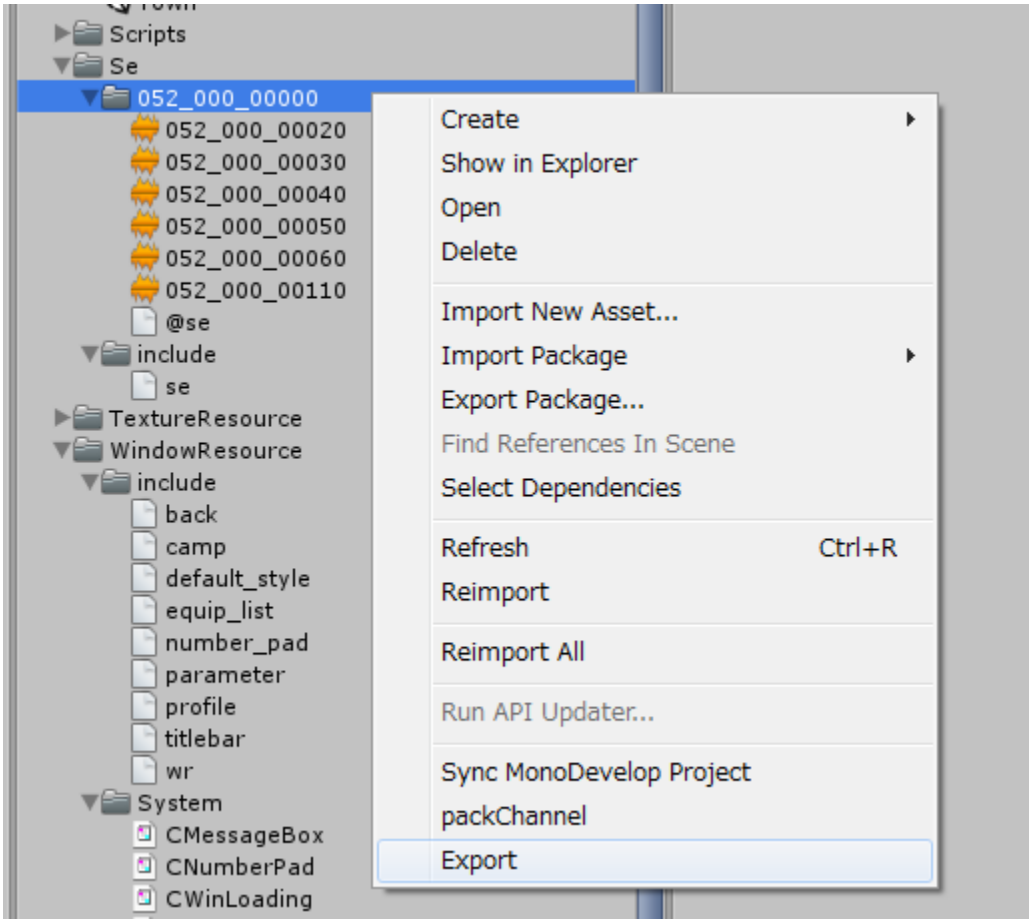
6.1.1 How to create a Sound Effect asset bundle

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_XXXXX.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;
```

(continues on next page)

(continued from previous page)

```
   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

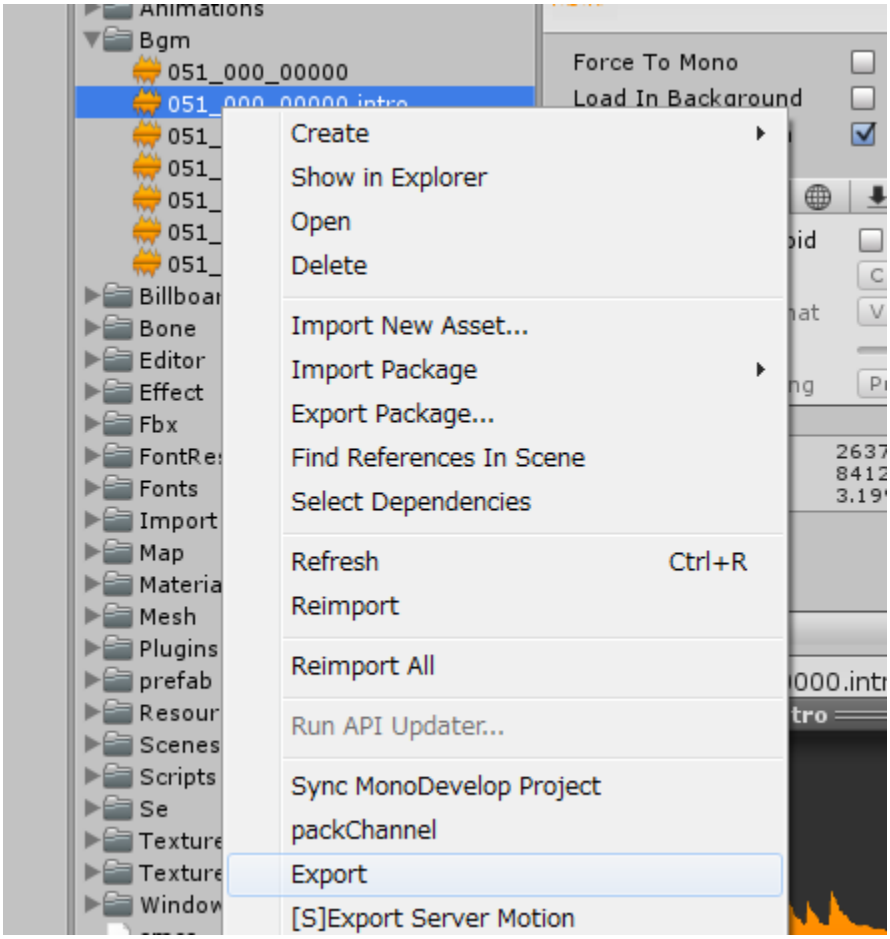
If necessary,BGM divide to the intro part and the loop part.You can create an asset bundle that were included them.The manager can be played automatically in order to read it.

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.



CHAPTER

SEVEN

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.

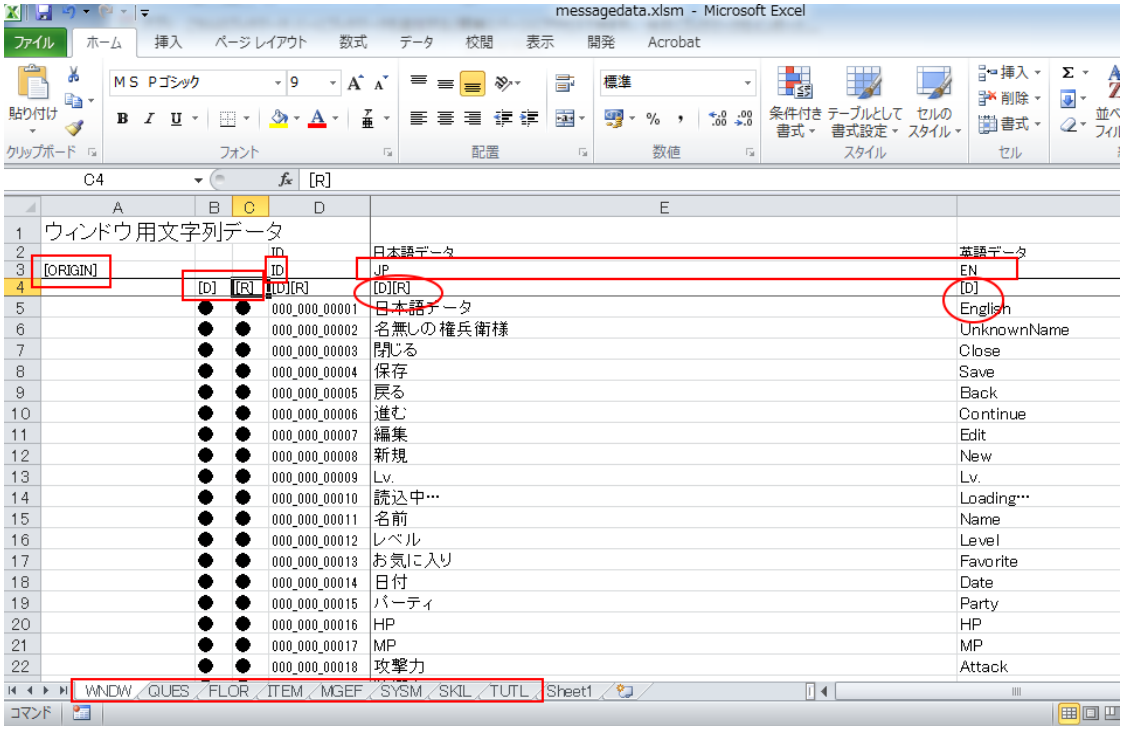
We have prepared the tool for outputting binary data for the resource data from the Excel data.

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

- `CMessageDataSheetMgr`
- `CMessageDataSheet`

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.

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 .

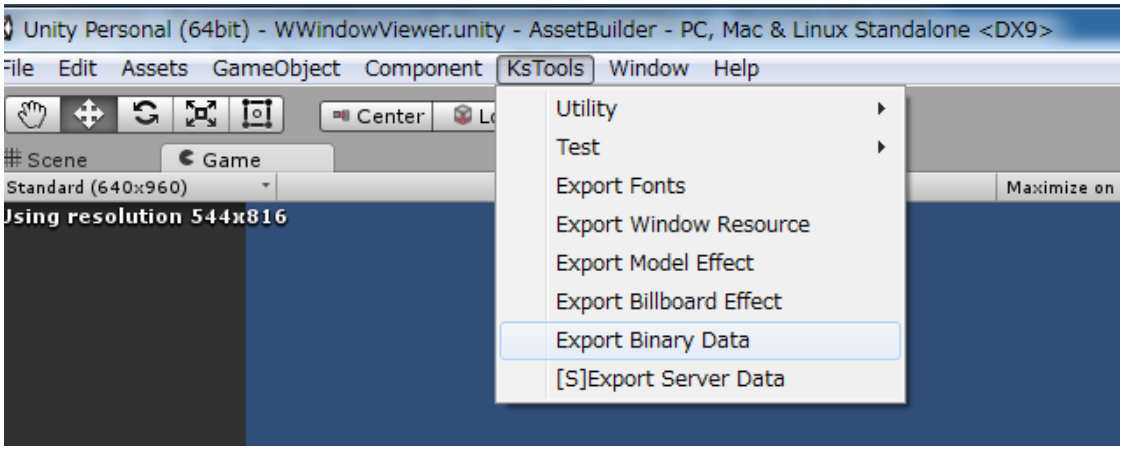
```
messagedata -v R messagedata.xlsx caption

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 [Tools]->[KsSoft]->[Export Binary Data] from the menu.



If you want to add the locale data, add in the *KsSoftConfig* .

CHAPTER

EIGHT

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_0000;
    STYLE = NOTITLEBAR|NOFRAME|ANCHOR_CENTER|TOP;
    SIZE = {100},{100};
    PRIORITY = PROGRESSBAR_PRIORITY;
};
METER(ProgressTotal) {
    ID = 000_001_00000;
    STYLE = ANCHOR_BOTTOM;
    SIZE = -128{100};
    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);
};
$y -=64;
METER(ProgressPart) {
   ID = 000_001_00010;
   STYLE = ANCHOR_BOTTOM;
   SIZE = -128{100},32;
   SIZE = -128{100},32;
   POSITION = 0,$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 = 0xfff123;      //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
0x12ffffff	Hexadecimal integer
000_001_00010	MultiID(format is 8bit,8bit,16bit)
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 *MulID*	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

CHAPTER

NINE

FEXIBLE COORDINATES, SIZING METHOD:

9.1 Percentages: flexible coordinates, sizing methods

The window system allows you to specify any percentage of the size of the parent when specifying almost any size, offset, and coordinates.

For example, you can specify that the width of a control is 75% of the window size.

This allows you to write windows without losing layout even if the screen size changes.

9.1.1 How to describe

Write the percentage by enclosing it in {...}. Values are percentages. You can also write expressions for offset and percent values.

Offset

Specifies only the offset.

The following example specifies a size of 40 x 40:.

```
SIZE = 20 * 2,16 + 24;
```

{percentage}

You can also specify a percentage only.

In the following example, if the screen size is 960 x 640, a size of 480 x 320 is specified.

```
SIZE = {50},{25 + 25};
```

offset + {percentage}

{percentage} + offset

Used to specify offsets and percentages.

In the following example, if the screen size is 960 x 640, a size of 520 x 360 is specified.

```
SIZE = 40 + {20 + 30},20 * 2{50};
   or
SIZE = {20 + 30} + 40,{50} + 20 * 2;
```

9.1.2 Wrong description example

You cannot include more than one percentage or offset.

```
SIZE = 10 + {20} + 30,40;

SIZE = 40 + {20} + {30},40;

SIZE = 40 + 0.5 * {20},40;
```

Both are incorrect descriptions.

9.2 How to set the virtual GUI screen size

It can be virtually fixed to a size specified by the developer, independent of the actual screen size.

If the specified horizontal size (or vertical size) is different from the actual screen size, all GUIs are automatically scaled.

This makes it relatively easy to handle mobile terminals with various screen sizes.

See *setUIResolution*.

CHAPTER

TEN

WINDOW:

10.1 An arrangement of the window

10.1.1 The screen:

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

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

You can use anchors to adjust the center or base position of a window.

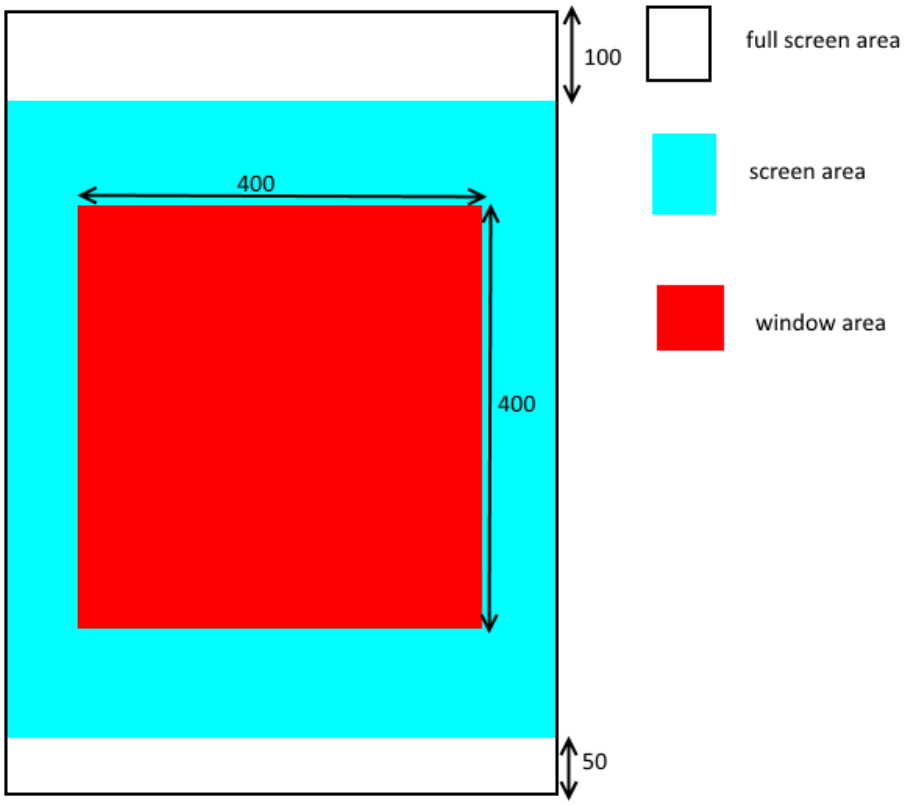
If you do not explicitly set it, the screen property size is the same as your device screen size.

```
SCREEN = 0,0,{100},{100};   //this mean is full screen.
```

The *percentage specification* allows you to specify a percentage of the screen size.

```
STYLE = ANCHOR_CENTER;
SCREEN = 0,100,0{100},-50{100};
SIZE = 400,400;
```

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



10.1.2 Anchor:

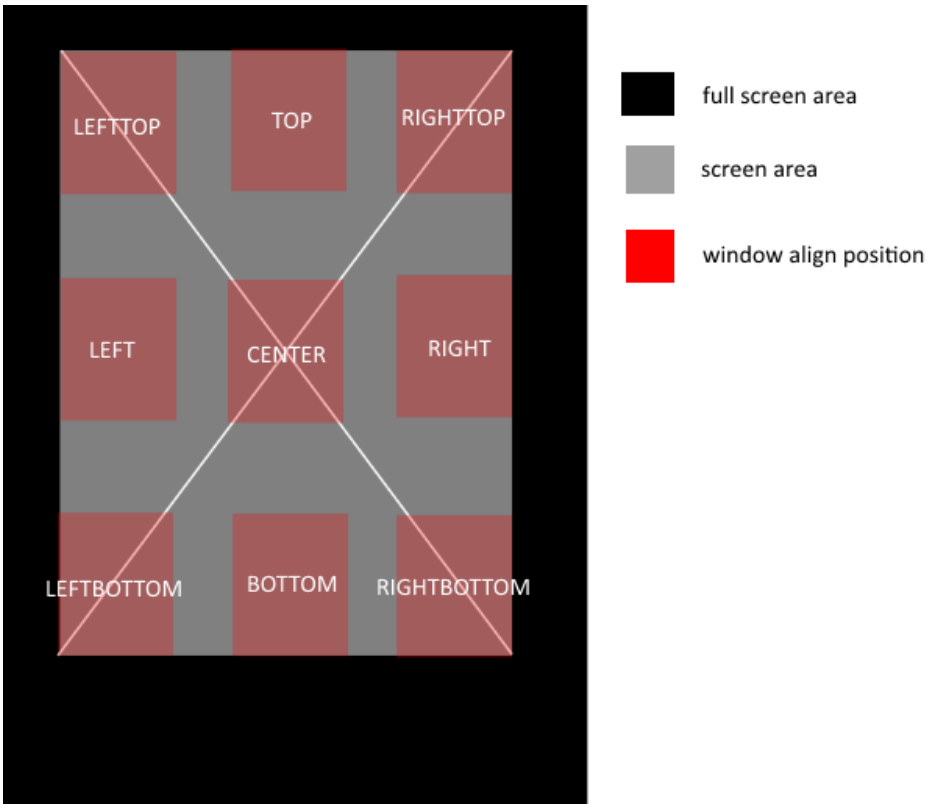
When placing the window, you can set individually the origin position on the screen.

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.



10.2 Window priority

Generally,the newly created window is displayed aboveb 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
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.
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.
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.
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)

10.3 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.

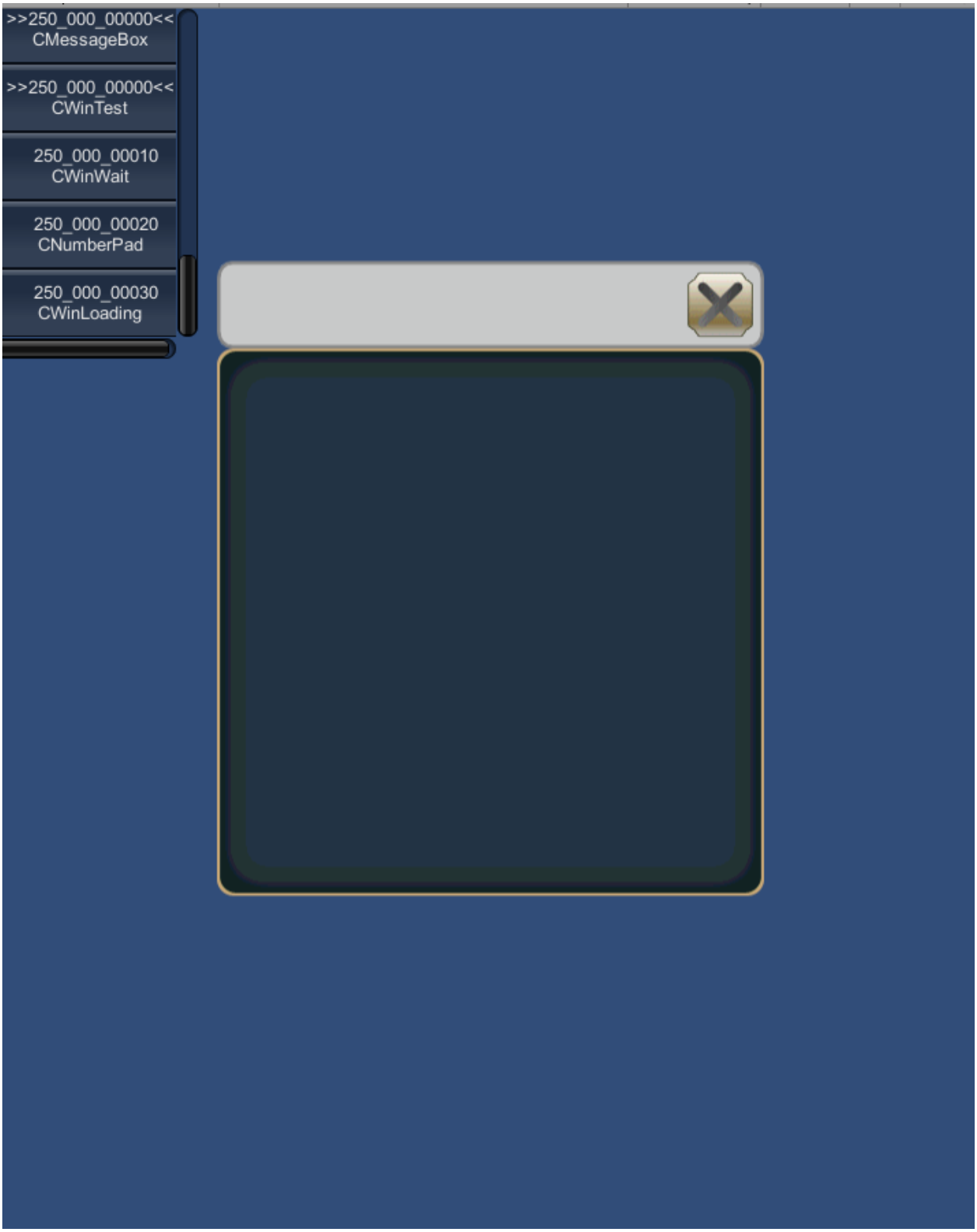
```
using UnityEngine;
using System;

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

10.3.1 Example

Minimum window script

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

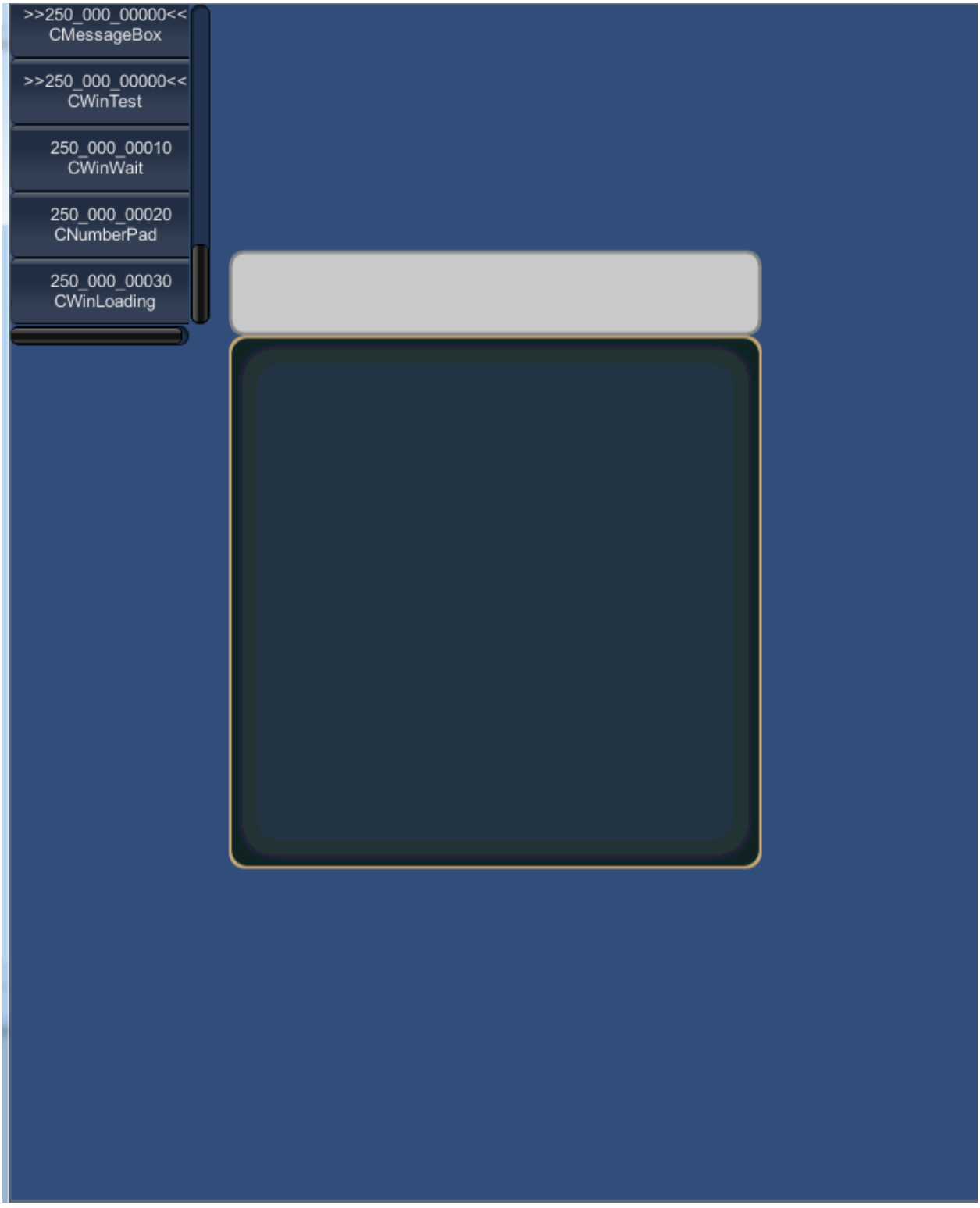
Title bar will appear when you described as follows.

```
WINDOW(250_000_00000) {
  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.

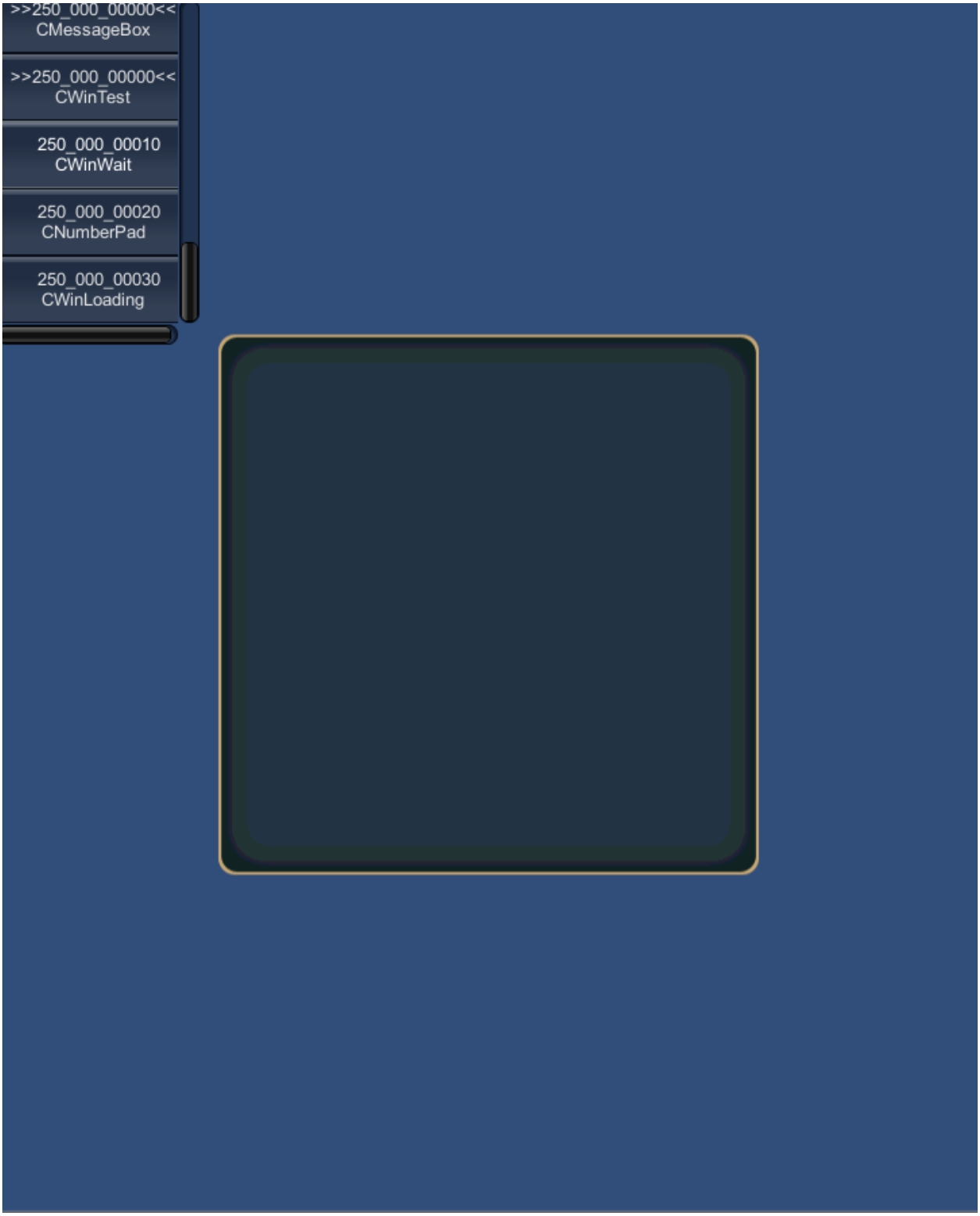
```
WINDOW(250_000_00000) {
  STYLE = NOCLOSE|ANCHOR_CENTER;
  SIZE = 400,400;
};
```



Window script without the title bar

Window script without the title bar

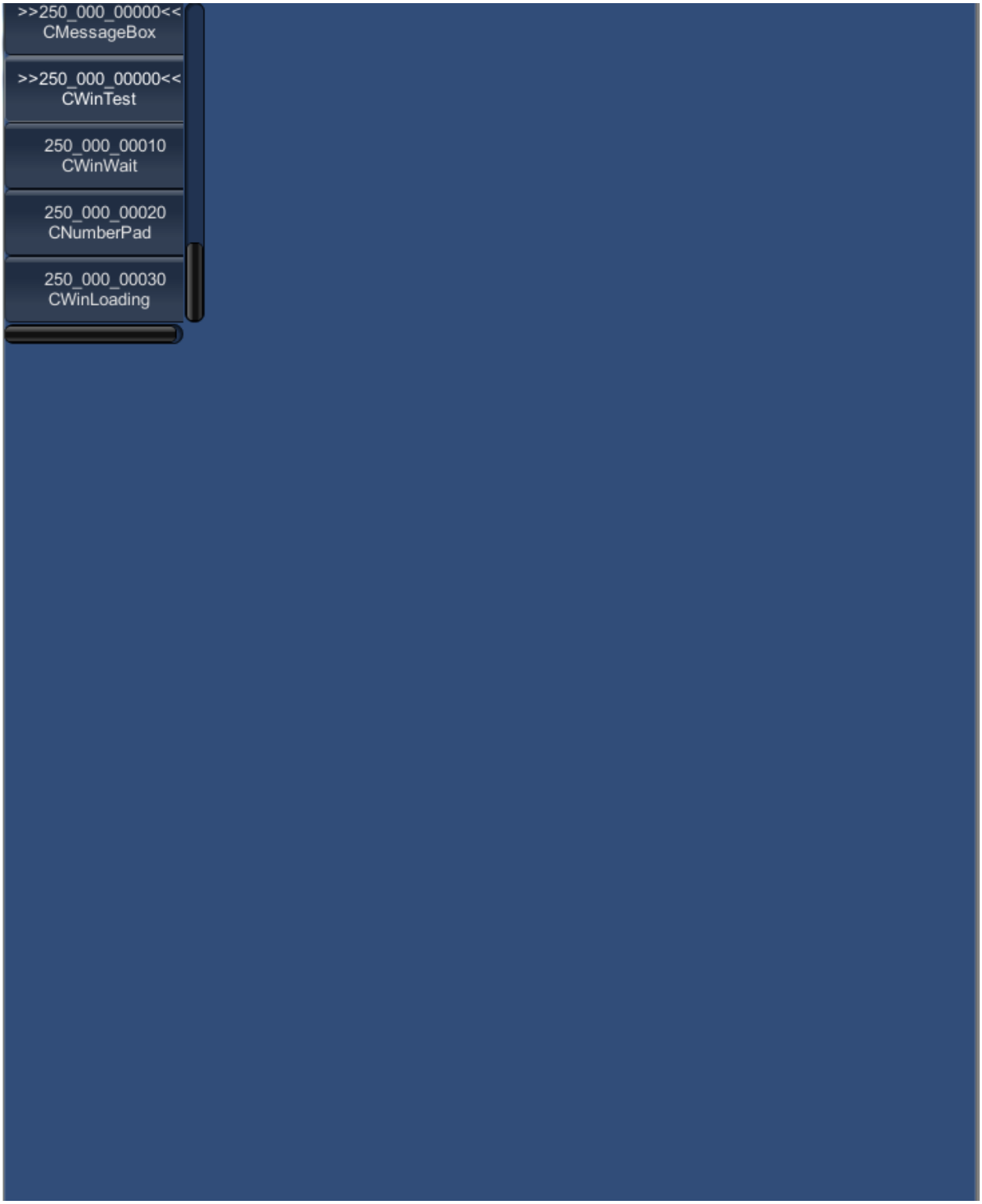
```
WINDOW(250_000_00000) {
  STYLE = NOTITLEBAR|ANCHOR_CENTER;
  SIZE = 400,400;
};
```



The window script which deleted all default indication

The window script which deleted all default indication

```
WINDOW(250_000_00000) {
  STYLE = NOTITLEBAR|NOFRAME|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 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 DRAG. Please try the following examples.

```
WINDOW(250_000_00000) {
  STYLE = ANCHOR_CENTER|DRAG|NOFRAME;
  POSITION = 0,0;
  CLOSE_POSITION = 400,0;
  SIZE = 400,400;
};
FRAME(Test) {
  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.

10.3.2 Property

RESOURCE = MulID

If you run a [Tools]->[KsSoft]->[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 resource data to the specified path.

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

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

In this example It outputs the resource 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 **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.

The coordinates can be expressed as a percentage from the *screen size* .

```
display position = (x,y) + screen size * ratio/100;
```

```
POSITOIN = 30{50},-40{50};
```

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

:: x = 30 + 640 * 50/100 = 350 y = -40 + 960 * 20/100 = 440

SIZE = width,height

Set the window size

The size can be a percentage of the *screen*.

```
display size = size + screen size * ratio/100;
```

```
SIZE = 100{50},50{20};
```

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

:: width = 100 + 640 * 50/100 = 420 height = 50 + 960 * 20/100 = 242

SCREEN = Top Left X,Top Left Y,width,height

Set the position and size of the screen.

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

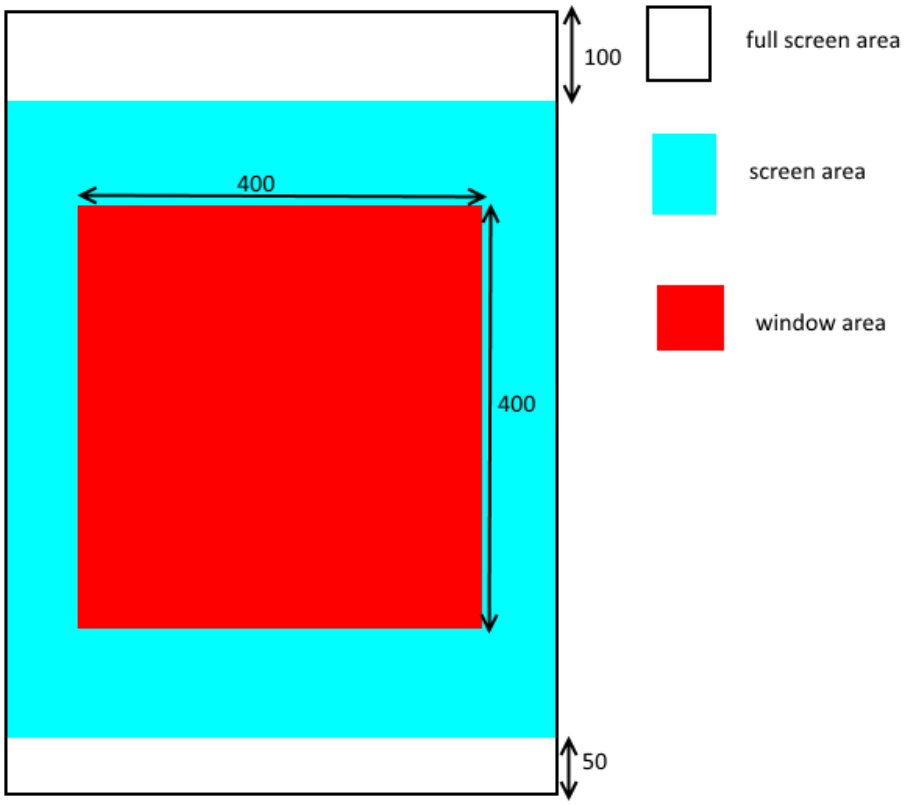
If you do not set the screen, the screen is set to full screen.

```
SCREEN = 0,0,0{100},0{100};      //Same as Full Screen
```

You can specify the percentage for all four parameters.

```
STYLE = ANCHOR_CENTER;
SCREEN = 0,100,0{100},-50{100};
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;     //Change to display on front
```

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

Display position anchor flag	Description
ANCHOR_DEFAULT	Set the anchor position in the upper left Same as ANCHOR_CENTER
ANCHOR_LEFTTOP	Set the anchor position in the upper left
ANCHOR_LEFT	Set the anchor position to the left. And vertical centering
ANCHOR_LEFTBOTTOM	Set the anchor position to the left. Located along the lower edge
ANCHOR_TOP	Set the anchor position to the upper side Centering at lateral.
ANCHOR_CENTER	Set the anchor position in the center of the window
ANCHOR_BOTTOM	Set the anchor position at the bottom Centering at lateral.
ANCHOR_RIGHTTOP	Set the anchor position in the upper right
ANCHOR_RIGHT	Set the anchor position to the right And vertical centering
ANCHOR_RIGHTBOTTOM	Set the anchor position to the right Located along the lower edge

Window display priority flag	Description
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.
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.
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.
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
NOCLOSE	Do not place the close button. When NOTITLEBAR is attached, close button is not placed
NOMINIMIZATION	Unimplemented
NOHELP	Unimplemented
NOTITLEBAR	Do not display the title bar.
NOFRAME	Do not display the frame.
DISABLE	Stop the function, can't accept input.
DRAG	Allow the window move with dragging.
NOACTIVE	It does not become active.
HIDE	Turn off the window display.
NOBRINGTOTOP	The display priority of a window doesn't allow to raise automatically when it becomes active.
OPENBOTTOM	Open the window on the back.

```
STYLE = NOTITLEBAR|NOFRAME;
```

CHAPTER

ELEVEN

CONTROL:

Common Specification:

11.1 An arrangement of the control

11.1.1 specifying coordinate ratios :

You can calculate the position of a control from *ratios* of its parent's size (often the window size).

```
POSITION = 0{20},50{30};
```

X = 0 + window width * 50/100

Y = 50 + window height * 30/100

11.1.2 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.

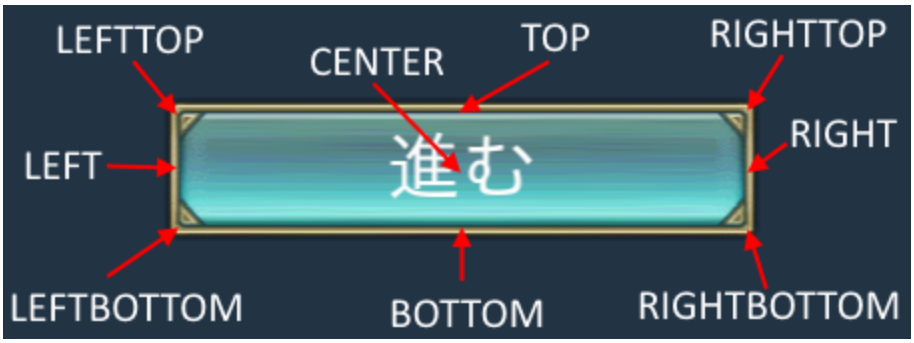
We call it the anchor in this window system.

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

11.1.3 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.



11.1.4 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

11.2 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

11.2.1 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

11.2.2 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 = POPUP|NOFRAME|NOTITLEBAR;
  TEXTURE_ZOFFSET = 100_100_00000,-1;
  TEXTURE_ZOFFSET = 100_010_00000,1;
  POSITION = 0,0;
  SCREEN = 0,0,{100},{-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: Meshes generated as render targets can be ordered with *TEXTURE_ZOFFSET* .See *RENDER* , *RENDERICON* .

11.3 The control size

11.3.1 size :ref:‘percentage<RATIOVALUE> specification ‘ :

You can calculate the size of a control from a *percentage* of the size of its parent (often the window size).

```
SIZE = -16 + {50},16 + {20};
```

In this example it is set as the following.

width = window width - 16

height = window height + 16

11.3.2 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.

11.4 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 .

11.4.1 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.

11.4.2 POSITION = X, Y

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

11.4.3 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
SIZE = {50} - 25;
```

Size can be specified as a *percentage* .

11.4.4 CONTENTS = { control define . . . }

It defines contents int the control.

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

11.4.5 CONTENTS_SIZE = width,height

It defines the size of the content.

```
CONTENTS_SIZE = {50},64;
```

Contents size can be specified as a *percentage* .

11.4.6 LINE_SPACE = line spacing pixel value

It defines the gaps between the content.

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

You cannot specify a percentage.

11.4.7 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;
```

11.4.8 PRIORITY = Priority value

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

```
PRIORITY = 3;
```

11.4.9 TEX_ID = Texture ID,Part ID

11.4.10 TEX_ID = Part ID

You set the texture ID and the part ID.

11.4.11 TEX_ID n = Texture ID,Part ID

11.4.12 TEX_ID n = Part ID

You set the texture ID and the part ID.

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

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

11.4.13 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 offset can be specified by a *percentage* .

11.4.14 Texture offset can be specified by a percentage .

You set the texture ID and the part ID.

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

When n = 0, it has 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
```

Texture size can be specified by a *percentage* .

11.4.15 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 .

11.4.16 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 set in the range of n = [1..7].

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

11.4.17 CAPTION = Caption ID

Set the ID of the string .

Please refer to *here* .

```
CAPTION = 010_000_00100;
```

11.4.18 CAPTION = “string”

Set the string.

```
CAPTION = "Hellow world!";
```

11.4.19 CAPTION_COLOR = R,G,B,A

You can set the caption color .

Set a value in the range of 0. . . 1.

11.4.20 CAPTION_OFFSET = X, Y

It adjust the caption position.

Caption offset can be specified by a *percentage* .

11.4.21 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.

11.4.22 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.

Base position change flag can adjust the position of a button,a label,and bar.

But it cann't adjust the position of *TEXT* , *RICHTEXT* or *LOGTEXT* .

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.
HIT	It is able to pressing the control.

Detailed specifications of each control:

11.5 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* .

You can not receive a callback when this control is pressed as default. If you want to receive any callbacks, you need to set HIT flag to STYLE property.

```
TEXT(Control name) {

  Property 1;

  Property 2;

     :

     :

  Property n;

};
```

11.5.1 Example

```
WINDOW(255_000_00001) {
  STYLE = NOTITLEBAR|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;
};
```

11.5.2 Property

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.

Note: This display position determines by display position anchor flag and Base position change flag.

A text anchor flag don't affects to the display positon.

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

Coordinates can be specified as a *percentage* .

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 = "Hellow 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.

Caption offset can be specified by a *percentage* .

CONTENTS_SIZE = Line feed size

When the string is longer than new line width , it will be a new line automatically.When it is not specified , it does not automatically line break.

```
CONTENTS_SIZE = 128;  //128pixel
```

The line feed size can be specified as a *percentage*.

LINE_SPACE = line spacing pixel value

Set the line spacing.

You cannot specify a percentage.

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

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.

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.
HIT	It is able to pressing the control.

11.6 RICHTEXT

C#: *CWinCtrlRichText*

It is a control for displaying rich text.

It is possible to change the text color , insert some texture part, and change the font type

RICHTEXT(Control name) {

Property 1;

Property 2;

:

:

Property n

};

11.6.1 Example

```
WINDOW(255_000_00001) {
  STYLE = NOTITLEBAR|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;
};
```

11.6.2 Property

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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

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 = "Hellow world!";
```

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

CAPTION_COLOR = R,G,B,A

You can set the caption color .

Set a value in the range of 0. . . 1.

CONTENTS_SIZE = Line feed size

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

```
CONTENTS_SIZE = 128;  //128pixel

CONTENTS_SIZE = {50} + 32;    // screen width * 0.5 + 32
```

The line feed size can be specified as a *percentage*.

LINE_SPACE = line spacing pixel value

Set the line spacing.

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

Line spacing cannot be specified as a percentage.

CAPTION_OFFSET = X, Y

It adjust the caption position.

Caption offset can be specified by a *percentage* .

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.
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.

```
"texture test\t[001_001_01000,icon0,32,32,0]\ndisplay test"
```

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 bule 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 *part* 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

When the specified caption string is touched,This window manager called *CWindowBase.onClick(CWinCtrlRichText cCtrl,CRichTextOne cText)* .

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.

If touch the texture, this window manager called *CWindowBase.onClick(CWinCtrlRichText cCtrl,CRichTextOne cText)* .

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 = "Texture test\t456[001_001_01000,icon0,32,32,0]";
```

11.7 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.

This control is scrollable control.

if you set *SCROLL_LOCK* to STYLE property, the window system calls onDrag when you drag these controls.And you can get a dragged direction of these controls.Even if you set DRAG to STYLE property, these controls do not affect anything.

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

LOG(Control name) {

Property 1;

Property 2;

:

:

Property n

};

11.7.1 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,-10;
};
```

11.7.2 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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

SIZE = width,height

Define the display area of the list box.

Controls outside the display area are clipped.

```
SIZE = {100} - 32,400;                  //display size
CONTENTS_SIZE = 800,80;                 //one content size
```

Size can be specified as a *percentage* .

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 = {100},40;       //virtual screen size

// Scroll to the left and right direction
SIZE = 400,400;                 //display size
CONTENTS_SIZE = {200},40;       //virtual screen size
```

Contents size can be specified as a *percentage* .

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) {
       :
  }
```

(continues on next page)

(continued from previous page)

```
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 **:func:'add a log<ADD_LOG>'** .

Note: The coordinates and the size *percentage* of the control in the content are calculated from the size of the list box.

Note that it is not a window size.

LINE_SPACE = line spacing pixel value

Set the line spacing.

You cannot specify a percentage.

```
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(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 .

STYLE = Flag0|Flag1|..|Flagn

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

Listbox control flag	Description
NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.
SCROLL_UNLOCK	Allow scrolling in list boxes.
SCROLL_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.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.
NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.
SCROLL_UNLOCK	Enable the scroll of contents.
SCROLL_LOCK	Disable the scroll of contents.

11.8 LOGTEXT

C#: *CWinCtrlLogText*

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

This behaviour is same as *TEXT* .

You can not receive a callback when this control is pressed as default. If you want to receive any callbacks, you need to set HIT flag to STYLE property.

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

LOGTEXT(Control name) {

Property 1;

Property 2;

:

:

Property n

};

11.8.1 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 = {
    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;
      LINE_SPACE = 2;
    };
  };
  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;
};
```

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

Note: This display position determines by display position anchor flag and Base position change flag.

A text anchor flag don't affects to the display positon.

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

Coordinates can be specified as a *percentage* .

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 = "Hellow 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.

Caption offset can be specified by a *percentage* .

CONTENTS_SIZE = Line feed size

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

```
CONTENTS_SIZE = 128;  //128pixel

CONTENTS_SIZE = {50} + 32;    // screen width * 0.5 + 32
```

The line feed size can be specified as a *percentage*.

LINE_SPACE = line spacing pixel value

Set the line spacing.

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

Line spacing cannot be specified as a percentage.

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.

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.
HIT	It is able to pressing the control.

11.9 EDITBOX

C#: *CWinCtrlEditbox*

It is a text editable control.

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

This control is scrollable control.

if you set *SCROLL_LOCK* to STYLE property, the window system calls onDrag when you drag these controls.And you can get a dragged direction of these controls.Even if you set DRAG to STYLE property, these controls do not affect anything.

EDITBOX(Control name)

Property 1;

Property 2;

:

:

Property n

};

11.9.1 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 = ANCHOR_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;
};
```

11.9.2 Editing features

This control have simple editing features on other than Android and iOS environment.

The editing shortcut keys are different from the UnityEditor and your application by the UnityEditor limitaion.

Cursor movement	On you application	On UnityEditor
Move the cursor to the right	Right	Right
Move the cursor to the left	Left	Left
Move the cursor to the up	Up	Up
Move the cursor to the down	Down	Down
Move the cursor to the top	Home	Home
Move the cursor to the end	End	End

Selected range.	On you application	On UnityEditor
Extend the selection to the right.	Shift + Right	Shift + Right
Extend the selection to the left.	Shift + Left	Shift + Left
Extend the selection to the up.	Shift + Up	Shift + Up
Extend the selection to the down.	Shift + Down	Shift + Down
Extend the selection to the top.	Shift + Home	Shift + Home
Extend the selection to the end.	Shift + End	Shift + End

Editing features	On you application	On UnityEditor
Back space	BackSpace	BackSpace
Delete	Delete	Delete
Cut	Ctrl + X	Ctrl + **Shift** + X
Paste	Ctrl + V	Ctrl + **Shift** + V
Cpopy	Ctrl + C	Ctrl + **Shift** + **Z**

11.9.3 Property

Default value

```
TEX_ID = "TXFD?";
TEX_ID1 = "CURSR"; //Cursor parts id
COLOR = 1,1,1,1;
COLOR1 = 1,1,1,1;  //Cursor color
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 = "TXFD?";
```

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

- TXFD0: Focus is in the state of OFF
- TXFD1: 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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

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
SIZE = {50} - 25;
```

Size can be specified as a *percentage* .

TEX_ID = Texture ID,Part ID

TEX_ID = Part ID

You set the texture ID and the part ID.

TEX_ID1 = Texture ID, part ID

TEX_ID1 = 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 .

LINE_SPACE = line spacing pixel value

It defines the gaps between the content.

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

You cannot specify a percentage.

GROUP = SCROLLBAR Control ID,. . .

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

It should be careful not to set to CONTENTS.

It is also possible to assign multiple scrollbars.

```
GROUP = SCROLLBAR(Horizon),SCROLLBAR(Vertical);
```

CAPTION = Caption ID

Set the ID of the string .

Please refer to *here* .

CAPTION = “string”

Set the string.

```
CAPTION = "Hellow world!";
```

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.

Caption offset can be specified by a *percentage* .

** 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.

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.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.
NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.
SCROLL_UNLOCK	Enable the scroll of contents.
SCROLL_LOCK	Disable the scroll of contents.

11.10 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.

This control is scrollable control.

if you set *SCROLL_LOCK* to STYLE property, the window system calls onDrag when you drag these controls.And you can get a dragged direction of these controls.Even if you set DRAG to STYLE property, these controls do not affect anything.

TEXTBOX(Control name) {

Property 1;

Property 2;

:

:

Property n

};

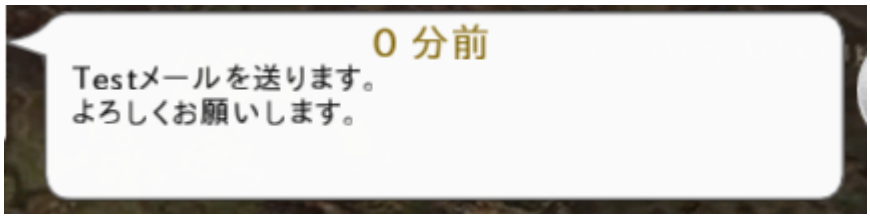
11.10.1 Example

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 = ANCHOR_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;
};
```



11.10.2 Property

```
TEX_ID = "TXFD?";
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 = "TXFD?";
```

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

- TXFD0: Focus is in the state of OFF
- TXFD1: 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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

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
SIZE = {50} - 25;
```

Size can be specified as a *percentage* .

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;
```

CAPTION = “string”

Set the string.

```
CAPTION = "Hellow world!";
```

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.

Caption offset can be specified by a *percentage* .

LINE_SPACE = line spacing pixel value

It defines the gaps between the content.

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

You cannot specify a percentage.

GROUP = SCROLLBAR Control ID,. . .

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

It should be careful not to set to CONTENTS.

It is also possible to assign multiple scrollbars.

```
GROUP = SCROLLBAR(Horizon),SCROLLBAR(Vertical);
```

** 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.

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.
DISABLE	It is not able to pressing the control. And the control color darken.
NOHIT	It is not able to pressing the control.
NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.
SCROLL_UNLOCK	Enable the scroll of contents.
SCROLL_LOCK	Disable the scroll of contents.

11.11 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

};

11.11.1 Example

```
WINDOW(255_000_00001) {
  STYLE = NOTITLEBAR|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;
};
```

11.11.2 Property

Default value

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

(continues on next page)

(continued from previous page)

```
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 = "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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

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
SIZE = {50} - 25;
```

Size can be specified as a *percentage* .

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 = "Hellow 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.

Caption offset can be specified by a *percentage* .

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 offset can be specified by a *percentage* .

Texture offset can be specified by a percentage .

You set the texture ID and the part ID.

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

When n = 0, it has 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
```

Texture size can be specified by a *percentage* .

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.

11.12 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

};

11.12.1 Example

```
WINDOW(255_000_00001) {
  STYLE = NOTITLEBAR|ANCHOR_CENTER;
  POSITION = 0,100;
  TEX_ID = 100_000_00001;
  CAPTION = 000_000_00010;
  SIZE = 512,256;
};
RADIO(Party) {
```

(continues on next page)

(continued from previous page)

```
  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.

11.12.2 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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

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
SIZE = {50} - 25;
```

Size can be specified as a *percentage* .

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 = "Hellow 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.

Caption offset can be specified by a *percentage* .

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 offset can be specified by a *percentage* .

Texture offset can be specified by a percentage .

You set the texture ID and the part ID.

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

When n = 0, it has 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
```

Texture size can be specified by a *percentage* .

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) {
```

(continues on next page)

(continued from previous page)

```
        :
    }
}
```

Method of accessing content using C # : *here*

Note: Unlike a list box, the coordinates and size *percentage specifications* for controls in the content are calculated from the window size.

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.

11.13 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

};

11.13.1 Example

```
WINDOW(255_000_00001) {
  STYLE = NOTITLEBAR|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 .

11.13.2 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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

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
SIZE = {50} - 25;
```

Size can be specified as a *percentage* .

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 = "Hellow 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.

Caption offset can be specified by a *percentage* .

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 offset can be specified by a *percentage* .

Texture offset can be specified by a percentage .

You set the texture ID and the part ID.

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

When n = 0, it has 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
```

Texture size can be specified by a *percentage* .

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*

Note: Unlike a list box, the coordinates and size *percentage specifications* for controls in the content are calculated from the window size.

** 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.

11.14 TEXTURE

C#: *CWinCtrlTexture*

This is a control for displaying a texture part .

It is possible to display eight texture parts.

You can not receive a callback when this control is pressed as default. If you want to receive any callbacks, you need to set HIT flag to STYLE property.

TEXTURE(Control name) {

Property 1;

Property 2;

:

:

Property n

};

11.14.1 Example

```
WINDOW(255_000_00001) {
  STYLE = NOTITLEBAR|ANCHOR_CENTER;
  POSITION = 0,100;
  TEX_ID = 100_000_00001;
  TEXTURE_ZOFFSET = 100_010_00000,1;
  SIZE = 512,256;
```

(continues on next page)

(continued from previous page)

```
};
TEXTURE(DecoLeft) {
  ID = 000_000_00040;
  STYLE = NOHIT;
  TEX_ID = 100_010_00000,"DECOL";
  POSITION = 24,24;
};
```

11.14.2 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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

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
SIZE = {50} - 25;
```

Size can be specified as a *percentage* .

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

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

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 offset can be specified by a *percentage* .

Texture offset can be specified by a percentage .

You set the texture ID and the part ID.

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

When n = 0, it has 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
```

Texture size can be specified by a *percentage* .

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 set in the range of n = [1..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.
HIT	It is able to pressing the control.

11.15 LINE

C#: *CWinCtrlLine*

This is the control to draw a line segment.

You can not receive a callback when this control is pressed as default. If you want to receive any callbacks, you need to set HIT flag to STYLE property.

LINE(Control name) {

Property 1;

Property 2;

:

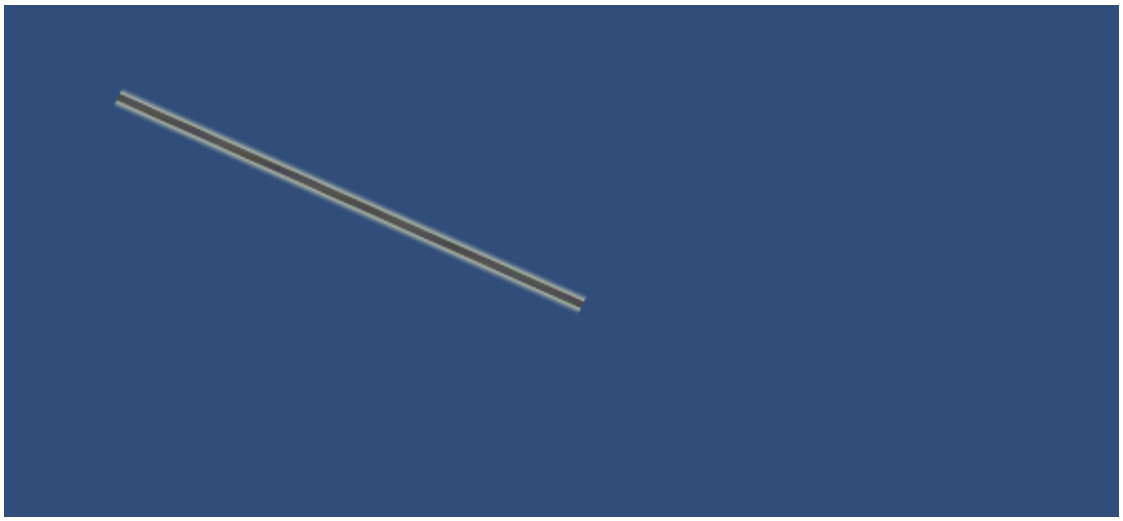
:

Property n

};

11.15.1 Example

```
WINDOW(255_000_00001) {
  STYLE = NOTITLEBAR|NOFRAME|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;
};
```



11.15.2 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

Determines the start coordinate. The reference position changes according to the STYLE *anchor* specification.

```
POSITION = 32,{50} + 64;
```

The start coordinates can be specified as a *percentage* .

POSITION2 = X, Y

Determines the end coordinate. The reference position changes according to the STYLE *anchor* specification.

```
POSITION2 = 32,{50} + 64;
```

The end coordinates can be specified as a *percentage* .

TEX_ID = Texture ID,Part ID

TEX_ID = Part ID

You set the texture ID and the part ID.

SIZE = Line Length, Line Weight

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

Set the thickness of the line segment in the Y.

Line weight can be specified as a *percentage* .

```
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.
DRAG	Enable a dragging.
DISABLE	It is not able to pressing the control. And the control color darken.
HIT	It is able to pressing the control.

11.16 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

};

11.16.1 Example

```
WINDOW(255_000_00001) {
  STYLE = NOTITLEBAR|ANCHOR_CENTER;
  POSITION = 0,100;
  SIZE = 512,256;
};

RENDER(AVATAR) {
  ID = 000_000_00200;
  STYLE = ANCHOR_CENTER;
  POSITION = 0,0;
  SIZE = {100}-64,{100}-64;
  CONTENTS_SIZE = 512,512;
};
```

11.16.2 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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

SIZE = width,height

Set the display size.

```
SIZE = 128,128;        //64x32
```

Size can be specified as a *percentage* .

Note: This property must always be set.

CONTENTS_SIZE = width,height

Set the display size.

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.

Note: It is possible to specify a percentage, but it is recommended to specify a straight value because of compatibility issues.

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.

11.17 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.

Unlike the *TEXTURE* control, it can have a caption.

ICON(Control name) {

Property 1;

Property 2;

:

:

Property n

};

11.17.1 Example

```
WINDOW(001_002_00003) {
  STYLE = NOTITLEBAR|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,"BNAD0";
  TEX_ID1 = 100_000_00001,"bd01";
  TEX_ID3 = 100_000_00001,"CRYS";
};
```

11.17.2 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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

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
SIZE = {50} - 25;
```

Size can be specified as a *percentage* .

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

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

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 offset can be specified by a *percentage* .

Texture offset can be specified by a percentage .

You set the texture ID and the part ID.

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

When n = 0, it has 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
```

Texture size can be specified by a *percentage* .

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 set in the range of n = [1..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 = "Hellow 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.

Caption offset can be specified by a *percentage* .

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.

11.18 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.

11.18.1 Example

```
WINDOW(255_000_00001) {
  STYLE = NOTITLEBAR|NOFRAME|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;
};
```

11.18.2 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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

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
SIZE = {50} - 25;
```

Size can be specified as a *percentage* .

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

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

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 offset can be specified by a *percentage* .

Texture offset can be specified by a percentage .

You set the texture ID and the part ID.

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

When n = 0, it has 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
```

Texture size can be specified by a *percentage* .

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 set in the range of n = [1..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 = "Hellow 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.

Caption offset can be specified by a *percentage* .

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.

11.19 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.

Like *ICON* , textures can also be rendered with it.

However, you should be careful with *TEXTURE_ZOFFSET* .

RENDERICON(Control name)

Property 1;

Property 2;

:

:

Property n

};

Note: Only one can be assigned to render.

11.19.1 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.

11.19.2 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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

TEX_ID = Texture ID

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: This property must always be set.

Note: Share the Render Texture only in the same window . Render Textures are not shared across a window.

Note: For each control, you can specify only one texture ID to render.

TEX_ID n = Texture ID,Part ID

TEX_ID n = Part ID

You set the texture ID and the part ID.

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

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

If you specify a Part ID, it is not rendered, but is rendered as a regular texture.

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: This property 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.

Note: It is possible to specify a percentage, but it is recommended to specify a straight value because of compatibility issues.

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 offset can be specified by a *percentage* .

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; //64x32
```

Size can be specified as a *percentage* .

Note: This property 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.

11.20 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* .

You can not receive a callback when this control is pressed as default. If you want to receive any callbacks, you need to set HIT flag to STYLE property.

METER(Control name) {

Property 1;

Property 2;

:

:

Property n

};

11.20.1 Example

```
WINDOW(255_000_00001) {
  STYLE = NOTITLEBAR|ANCHOR_CENTER;
  POSITION = 0,100;
  TEX_ID = 100_000_00001;
  CAPTION = 000_000_00010;
  SIZE = 512,256;
};
METER(Progress) {
  ID = 000_001_00000;
  STYLE = ANCHOR_CENTER;
  SIZE = {100}-128;
  POSITION = 0,0;
  TEX_ID = 0,"MTR";
  COLOR = 1,1,1,1;
  TEX_ID1 = 0,"MTRB";
  COLOR1 = 1,1,1,1;
};
```

11.20.2 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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

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
SIZE = {50} - 25;
```

Size can be specified as a *percentage* .

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

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

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 offset can be specified by a *percentage* .

Texture offset can be specified by a percentage .

You set the texture ID and the part ID.

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

When n = 0, it has 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
```

Texture size can be specified by a *percentage* .

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 set in the range of n = [1..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.
HIT	It is able to pressing the control.

11.21 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*

You can not set HIT or NOHIT flag to STYLE property.

SCROLLBAR(Control name) {

Property 1;

Property 2;

:

:

Property n

};

11.21.1 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;
};
```

11.21.2 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 = NOTITLEBAR|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;
```

(continues on next page)

(continued from previous page)

```
                        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|ITEM_STACK_H;
        TEX_ID = "SCBBV";
        POSITION = 0,0;
        SIZE = 0,0;
};
SCROLLBAR(BOTTOM) {
        STYLE = ANCHOR_LEFTBOTTOM|ITEM_STACK_H;
        TEX_ID = "SCBBV";
        POSITION = 0,0;
        SIZE = 0,0;
};
```

11.21.3 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: ITEM_STACK_V
- Horizontal: ITEM_STACK_H

11.21.4 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.

11.21.5 Property

Default value

```
TEX_ID = "SCBRH";
STYLE = 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;
```

You cannot specify a percentage.

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
```

You cannot specify a percentage.

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.
ITEM_STACK_V	Set to the direction of the scroll bar in the vertical direction.
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.

11.22 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 .

This control is scrollable control.

if you set *SCROLL_LOCK* to STYLE property, the window system calls onDrag when you drag these controls.And you can get a dragged direction of these controls.Even if you set DRAG to STYLE property, these controls do not affect anything.

LISTBOX(Control name) {

Property 1;

Property 2;

:

:

Property n

};

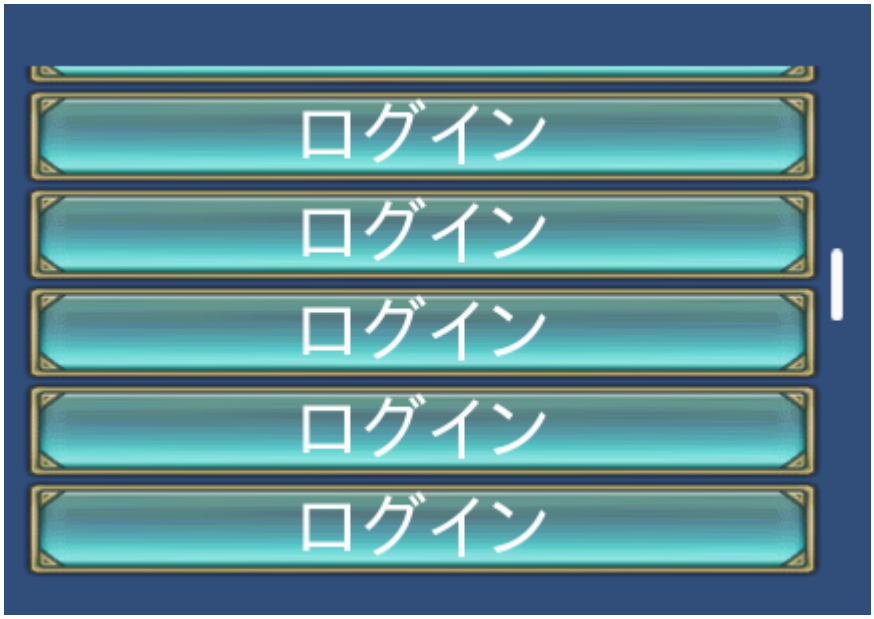
11.22.1 Example

```
WINDOW(255_000_00001) {
  STYLE = NOTITLEBAR|NOFRAME|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 = {
```

(continues on next page)

(continued from previous page)

```
    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;
};
```



11.22.2 Property

Default value

```
STYLE = 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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

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 = 512,512;        //512x512
SIZE = {50} - 100,{50} + 100; //screen width * 0.5 - 100,screen height * 0.5
↪+ 100
```

Size can be specified as a *percentage* .

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 size can be specified as a *percentage* .

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) {
```

(continues on next page)

(continued from previous page)

```
        :
    };
    CHECKBOX(B) {
        :
    };
    LISTBOX(C) {
        :
    };
```

How to access to a particular control of the content list from C#, please refer to *here*.

Note: The coordinates and the size *percentage* of the control in the content are calculated from the size of the list box.

Note that it is not a window size.

LINE_SPACE = line spacing pixel value

It defines the gaps between the content.

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

You cannot specify a percentage.

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
ITEM_STACK_V	List box arranged in the vertical direction.
ITEM_STACK_H	List box arranged in the vertical direction.
NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.
SCROLL_UNLOCK	Enable the scroll of contents.
SCROLL_LOCK	Disable the scroll of contents.

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.
ITEM_STACK_V	List box arranged in the vertical direction.
ITEM_STACK_H	List box arranged in the vertical direction.
NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.
SCROLL_UNLOCK	Enable the scroll of contents.
SCROLL_LOCK	Disable the scroll of contents.

11.23 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.

This control is scrollable control.

if you set *SCROLL_LOCK* to STYLE property, the window system calls onDrag when you drag these controls.And you can get a dragged direction of these controls.Even if you set DRAG to STYLE property, these controls do not affect anything.

LISTBOXEX(Control name) {

Property 1;

Property 2;

:

:

Property n

};

11.23.1 Example

```
WINDOW(255_000_00001) {
  STYLE = NOTITLEBAR|NOFRAME|ANCHOR_CENTER;
  POSITION = 0,100;
  SIZE = 512,256;
};

$w = 400;
LISTBOXEX(List) {
  ID = 001_100_00000;
  POSITION = 0,0;
  STYLE = ANCHOR_BOTTOM;
  SIZE = $w,{100} - 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 = 4,16;
  SIZE = 0,-16 * 2;
};
```

11.23.2 Property

Default value

```
STYLE = 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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

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 = 512,512;        //512x512
SIZE = {50} - 100,{50} + 100; //screen width * 0.5 - 100,screen height * 0.5
↪+ 100
```

(continues on next page)

(continued from previous page)

Size can be specified as a *percentage* .

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 size can be specified as a *percentage* .

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*.

Note: The coordinates and the size *percentage* of the control in the content are calculated from the size of the list box.

Note that it is not a window size.

LINE_SPACE = line spacing pixel value

It defines the gaps between the content.

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

You cannot specify a percentage.

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
ITEM_STACK_V	List box arranged in the vertical direction.
ITEM_STACK_H	List box arranged in the vertical direction.
NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.
SCROLL_UNLOCK	Enable the scroll of contents.
SCROLL_LOCK	Disable the scroll of contents.

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.
ITEM_STACK_V	List box arranged in the vertical direction.
ITEM_STACK_H	List box arranged in the vertical direction.
NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.
SCROLL_UNLOCK	Enable the scroll of contents.
SCROLL_LOCK	Disable the scroll of contents.

11.24 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.

This control is scrollable control.

if you set *SCROLL_LOCK* to STYLE property, the window system calls onDrag when you drag these controls.And you can get a dragged direction of these controls.Even if you set DRAG to STYLE property, these controls do not affect anything.

CONTAINER(Control name) {

Property 1;

Property 2;

:

:

Property n

};

11.24.1 Example

```
WINDOW(255_000_00001) {
  STYLE = NOTITLEBAR|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 = {100},{100};
```

(continues on next page)

(continued from previous page)

```
  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";
    };
  };
};
```

11.24.2 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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

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 = 512,512;        //512x512
SIZE = {50} - 100,{50} + 100; //screen width * 0.5 - 100,screen height * 0.5␣
↪+ 100
```

Size can be specified as a *percentage* .

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 size can be specified as a *percentage* .

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*

Note: The coordinate and size *ratio* specifications for controls in the content are calculated relative to the size of the container.

Note that it is not a window 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 .

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
NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.
SCROLL_UNLOCK	Enable the scroll of contents.
SCROLL_LOCK	Disable the scroll of contents.

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.
NOBOUNCES	When you've stopped it to scroll ,it is stopped even if there are still the momentum.
SCROLL_UNLOCK	Enable the scroll of contents.
SCROLL_LOCK	Disable the scroll of contents.

11.25 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;

:

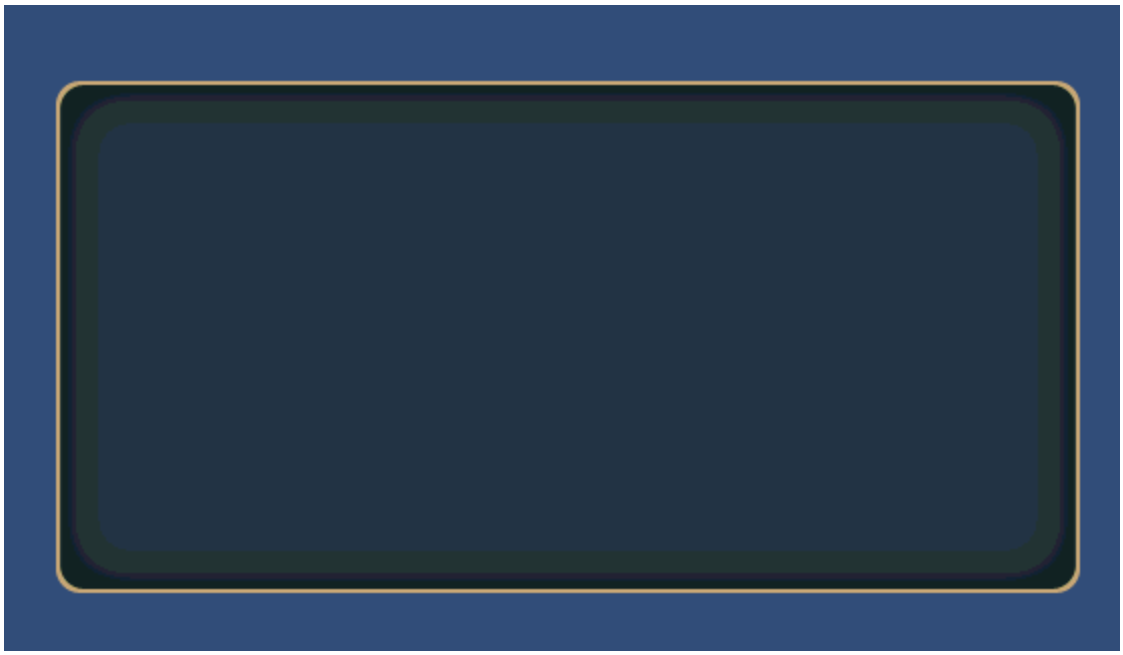
:

Property n

};

11.25.1 Example

```
WINDOW(255_000_00001) {
  STYLE = NOTITLEBAR|NOFRAME|ANCHOR_CENTER;
  POSITION = 0,100;
  SIZE = 512,256;
};
FRAME(Test) {
  ID = 000_000_00010;
  POSITION = 0,0;
  SIZE = RELATIVE_SIZE(0),RELATIVE_SIZE(0);
};
```



11.25.2 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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

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
SIZE = {50} - 25;
```

Size can be specified as a *percentage* .

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

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.

11.26 LABEL

C#: *CWinCtrlLabel*

This is the control to display a label.

You can not receive a callback when this control is pressed as default. If you want to receive any callbacks, you need to set HIT flag to STYLE property.

LABEL(Control name) {

Property 1;

Property 2;

:

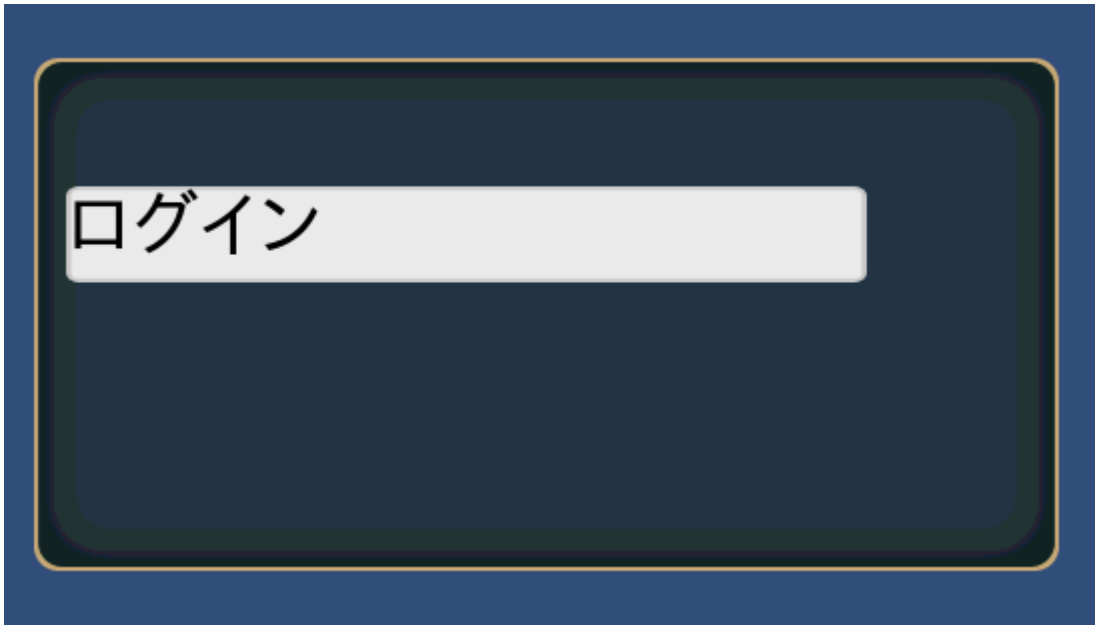
:

Property n

};

11.26.1 Example

```
WINDOW(255_000_00001) {
  STYLE = NOTITLEBAR|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;
};
```



11.26.2 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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

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
SIZE = {50} - 25;
```

Size can be specified as a *percentage* .

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 = "Hellow 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.

Caption offset can be specified by a *percentage* .

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.
HIT	It is able to pressing the control.

11.27 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
};
```

11.27.1 Example

```
WINDOW(255_000_00001) {
  STYLE = NOTITLEBAR|ANCHOR_CENTER;
  POSITION = 0,100;
  SIZE = 512,256;
};
```

(continues on next page)

(continued from previous page)

```
$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;
};
```

11.27.2 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

Determine the display position. The base position changes according to the *anchor* in STYLE.

```
POSITION = 32,{50} + 64;
```

Coordinates can be specified as a *percentage* .

SIZE = width,height

Define the display area of the list box.

Controls outside the display area are clipped.

```
SIZE = {100} - 32,400;                  //display size
CONTENTS_SIZE = 800,80;                 //one content size
```

Size can be specified as a *percentage* .

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 = "Hellow 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.

Caption offset can be specified by a *percentage* .

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:

11.28 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 .

class CMainSystemBase

void **Awake** ()

By registering various managers in the Awake function using AddComponent and addManager, the Start function automatically performs necessary initialization and loading of necessary asset bundles.

A typical description 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;
```

(continues on next page)

(continued from previous page)

```
    // Add Component
    gameObject.AddComponent<CInput>();
    if (KsSoftConfig.UseAssetBundle) {
        gameObject.AddComponent<CAssetBundleMgr>();
    }
    gameObject.AddComponent<CSpriteFontMgr>();
    gameObject.AddComponent<CTextureResourceMgr>();
    gameObject.AddComponent<CWindowMgr>();
    gameObject.AddComponent<CBgmResourceMgr>();
    gameObject.AddComponent<CSeResourceMgr>();
    gameObject.AddComponent<CSoundEffectMgr>();

    // Managers that need to be initialized.
    addManager(new CMessageDataSheetMgr(Utility.getSystemLocale()));
}
```

void **addManager** (*IManager mgr*)

> **Param IManager mgr** It is an interface required for the objects that must be initialized with the loaded asset bundle.

Note: Within Awake, you need to AddComponet.

---

virtual void **initialize** ()

As soon as the asset bundle is loaded, the initialization process is called through the interfaces: ILoader, IManager.

virtual void **OnChangeScene** ()

Called when you switch scenes.

bool **changeScene** (string *sScene*, bool *bForceChange=false*)

Switches to the specified scene.

Equivalent to Application.LoadLevel, but differs in the following ways:

1. Wait for the fade object to finish fading.
2. While loading the asset bundle, the scene switch is pending.
3. Automatically closes the created window.

OnChageScene is called just before you switch scenes (Just before Application.LoadLevel is called).

bool **isChangingScene** { **get;** }

Determines whether a scene is being imported.

Available only when switching scenes via CMainSystemBase.chageScene.

bool **isInitialized** { **get;** }

Determines whether initialization is complete.

static int **randi** ()

Returns a random integer.

static int **randi** (int *iMax*)

Returns a random number in the specified range.

static int **randi** (int *iMin*, int *iMax*)

Returns a random number in the specified range.

static float **randf** ()

Returns a floating point random number.

class ILoader

bool **isLoading** ()

Return bool true while loading, false when finished loading

Follow the script that inherits MonoBehaviour.

In a script in the same game object as CMainSystemBase, examine the object that has the ILoader, and wait until ILoader.isLoading () is false.

Note: Within Awake, you need to AddComponet.

---

class IManager

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());
```

bool **initialize** (*CAssetBundle*[] *aAssetBundles*)

Param CAssetBundle[] aAssetBundles array of loaded asset bundles

void **release** ()

Release process called when the game object is released

uint[] **getAssetBundleIds** ()

Return uint[] ID of the asset bundle that needs to be loaded at initialization

This function returns an array of the asset bundle IDs that need to be loaded.

11.29 CAssetBundleMgr

class 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.

const int **MaxConcurrentLoadNum** { **get; set;** }

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** { **get;** }

It sets the maximum cache size of a local disk.

The default value is set to 512 * 1024 * 1024 (512Mbyte).

static string **httpServerPath** { **get; set;** }

It sets the HTTP address.

static string **debugPath** { **get; set;** }

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** { **get;** }

It has the error number

int **errorMessage** { **get;** }

It has the error message

CAssetBundle[] **lodings**

This returns a list of asset bundles currently being downloaded .

This list includes what the manager deferred assetbundle downloads.

int **loadNum** { **get;** }

It has a number of asset bundle currently being downloaded .

This number includes what the manager deferred assetbundle downloads.

static *CAssetBundleMgr* **Instance** { **get;** }

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.

class CAssetBundle

It can be cast in AssetBundle.

AssetBundle **get** ()

Get the AssetBundle.

uint **id** { **get;** }

It has the ID.

bool **isLoaded** { **get;** }

if it is already loaded,this value sets to true.

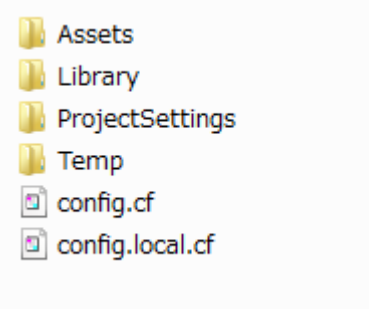
You can see whether it is resident .

11.29.1 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
     if (KsSoftConfig.UseAssetBundle) {
             gameObject.AddComponent<CAssetBundleMgr>();
         }
```

11.30 About asset bundle

"CAssetBundleMgr" first loads "version.unit3d".This section describes how this "version.unity3d" works.

11.30.1 When "version.unity3d" is updated

- [Tools]->[KsSoft]->[Export Binary Data]
- [Tools]->[KsSoft]->[Export Window Resource]

Automatically updates when you create or update any asset bundle.

Within these tools, the following functions are called.

```
ExportVersion.export(BuildTarget eTarget);
```

"version.unity3d" contains the following information before export.

- Asset bundles that existed
- MD5 of each asset bundle
- Version number of each asset bundle
- Information on which asset bundles are resident after loading

By calling this function, you are rebuilding "version.unity3d" if any changes were made compared to MD5 in the asset bundle in the folder corresponding to the build target.

At this time, the version number generated from the current time is given to the updated file.

This allows you to simply load "version.unity3d" to:

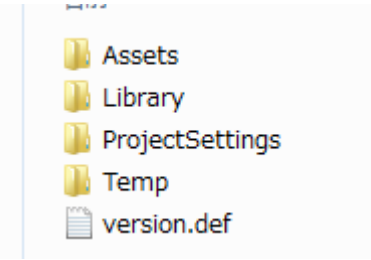
- Know what's in the asset bundle ahead of time.
- "LoadFromCacheOrDownload" argument, version, can only be newly allocated to files that have changed, so only updates can be downloaded.

Note: If you want to create your own asset bundle, remember to export "version.unity3d" at the end of the script.

11.30.2 Asset Bundle Residency

It is possible to set the downloaded asset to reside during the "version.unity3d" update.

First, create a file named "version.def" in the same location as "Assets" folder.




```
#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;
```

The asset bundle you specify will reside after loading.

11.30.3 version.txt

"version.txt" is a "version.unity3d" that was visualized for debugging purposes only.

Unnecessary file at export/runtime.

11.30.4 Collisions, such as with version control tools

Please follow the steps below.

When asset bundles themselves collide

Please try to solve it.

When "version.unity3d" collides

Remove "version.unit3d" once after resolving all asset bundle conflicts.It then download the uploaded "version.unit3d".Finally, re-export "version.unit3d" in the local environment in the following way:.

- [Tools]->[KsSoft]->[Re-Export Version]

11.31 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 .

---

class CTextureResourceMgr

CTextureResource **reference**（uint *id*）

Param uint id Asset Bundle ID

Refers to a texture resource. If not cached, download it.

Returns CTextureResource if the texture resource exists even if it has not already been downloaded.

The returned CTextureResource is being downloaded can be determined by CTextureResource.isLoaded.

if it returns true,the asset is already loaded.

If it cannot be downloaded as an asset bundle, try to import it as resource data.

If none exists, null is returned.

static CTextureResourceMgr **Instance** { **get;** }
It has an instance of a manager

class CTextureResource
This can be cast to the Material.

Material **get**（）
You get the Material .

uint **id** { **get;** }
It has the ID.

bool **isLoaded** { **get;** }
if it is already loaded,this value sets to true.

bool **isRemain** { **get;** }
You can see whether it is resident .

CSpriteDataOne[] **spriteData** { **get;** }
You get the sprite data .

Parts of UV information, patch information , and color information are stored.

class CSpriteDataOne
Parts of UV information, patch information , and color information are stored.

uint **m_id** { **get;** }

WinColor **m_color** { **get;** }

e_Patch **m_ePatch** { **get;** }

```
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** { **get; set;** }

UV texture information are stored .

Vector4[] **m_aUV** { **get; set;** }

If e_Patch is H3 or V3, it is stored three UV information. If e_Patch is HV9, it is stored nine UV information .

11.32 CSoundEffectMgr

t has multiple AudioSources to group, prioritize, and limit the number of simultaneous sound effects.

This is the default concurrency limit.

- 2D Audio: 4
- 3D Audio: 8

You can also set individual limits for the number of simultaneous sounds per group.

Find out how to configure these settings for SE *here* .

This manages AudioSources, eliminating the need for programmers to create their own AudioSources.

11.32.1 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>();
        if (KsSoftConfig.UseAssetBundle) {
```

(continues on next page)

(continued from previous page)

```
                gameObject.AddComponent<CAssetBundleMgr>();
            }
        gameObject.AddComponent<CSpriteFontMgr>();
        gameObject.AddComponent<CTextureResourceMgr>();
        gameObject.AddComponent<CWindowMgr>();
        gameObject.AddComponent<CBgmResourceMgr>();
        gameObject.AddComponent<CSeResourceMgr>();
        gameObject.AddComponent<CSoundEffectMgr>();

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

class CSoundEffectMgr

void **playBgm** (*CBgmResource cBgmResource*)

Param CBgmResource Imported BGM Resource

void **playBgm** (uint *id*)

Param uint id Asset Bundle ID

Plays the specified BGM. If the specified asset bundle is not loaded, playback will start automatically after loading.If you want to explicitly wait for loading, wait until cBgmResource.isLoaded () returns true before playing.

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*)

Param CSoundEffect cSE The SE object available from CSeResource.

Plays the specified SE as 2D audio. You can get a CSoundEffect in the following ways:.

```
CSeResource   m_cSR;
          :
CSoundEffect  cSE = m_cSR.find(id);
```

void **play** (uint *mAssetBundle*, uint *id*)

Param uint mAssetBundle Asset Bundle ID

Param uint id SE ID

Plays the specified SE as 2D audio.

void **play** (CSoundEffect *cSE*, IEffectEmitter *emitter*)

void **play**（CSoundEffect *cSE*, Transform *trans*）
void **play**（CSoundEffect *cSE*, Vector3 *position*）

Param CSoundEffect cSE The SE object available from CSeResource.

Param IEffectEmitter emitter Emitters that produce the effect.

Param Transform trans Where the sound is coming from

Param Vector3 position Where the sound is coming from

Plays the specified SE as 3D audio.

It has an instance of a manager

class CSeResourceMgr

It is the container that manages the CSeResource.

After downloading asset bundles via CassetBundleMgr, it manages the imported audio data.

CSeResource **refenrece**（uint *mAssetBundle*）

Param uint mAssetBundle Specifies the SE Resource ID(Asset Bundle ID).

Return The SE resource is returned.

Get SE resources. The assets you can get are converted *here* .

CSoundEffect **find**（uint *mAssetBundle*, uint *mId*）

Param uint mAssetBundle Specify the SE resource.

Param uint id Specify the SE ID.

Return SE will be returned. If the asset bundle that the specified SE is packed in is still being loaded or does not exist, null is returned.

Gets a CSoundEffect.

static CSeResourceMgr **Instance** { **get;** }

It has an instance of a manager

class CSeResource

It keeps the asset-bundled SE available.

There are multiple SEs in one CSeResource.

It has an IWinSoundEffect interface and can be registered with CWindowMgr.

you can play the SE through the CSoundEffectMgr.

CSoundEffect **find**（uint *id*）

Param uint id Specifies the SE ID you want to retrieve.

Return Return the SE.

SE will be returned.

void **play**（uint *mSE*）

Param uint id Specifies the SE ID you want to play.

Play SE.

bool **isLoaded** { **get;** }

Check to see if the asset bundle has finished loading.

When true, the load is complete and ready for use.

class CBgmResourceMgr

This is the container that manages the CBgmResource.

It manages the BGM resources that have downloaded the asset bundles through CAssetBundleMgr.

To sound the BGM, use CSoundEffectMgr.

CBgmResource **refenrece** (uint *mAssetBundle*)

Param uint mAssetBundle Specifies the BGM resource ID (Asset Bundle ID).

Return BGM resource will be returned.

Gets the BGM resource. The assets you can get are converted *here* .

CBgmResource **load** (uint *mResource*)

Param uint mResource Specifies the resource ID.

Return BGM resource will be returned.

You can import a file by placing it under the Resources folder as follows:

Assets/Resources/001_000_000000.mp3 Assets/Resources/001_000_000000.intro.mp3

static CBgmResourceMgr **Instance** { **get;** }

It has an instance of a manager

class CBgmResource

The asset-bundled BGM is kept available.

Within the CBgmResource, there is audio data for the intro and loop.

To sound the BGM, use CSoundEffectMgr.

AudioClip **loopClip** { **get;** }

Clip in the loop

AudioClip **introClip** { **get;** }

Clip in the intro

bool **isLoaded** { **get;** }

Check to see if the asset bundle has finished loading.

When true, the load is complete and ready for use.

11.33 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 .

11.33.1 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<CAssetBundleMgr>();
        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 .

class 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*)

Param byte[] buffer Binary data read

Specifies binary data generated from Excel.

CMessageDataSheet **find** (uint *sheetname*)

Param uint type Pass the sheet name with *FiveCC*.

Gets one sheet.

```
CMessageDataSheet cSheet = cMessageDataSheetMgr.find(new FiveCC("WNDW");
```

Get the currently set locale .

Gets an instance of CMessageDataSheetMgr.

class CMessageDataSheet

It corresponds to the one sheet on an Excel file.

string **find** (uint *id*)

Param uint id Specifies *MultiID* and retrieves text resources.

string **format** (uint *id*, *params* object[] *args*)

Param uint id Specifies *MultiID* and retrieves text resources.

Param params object[] args variable argument

Generates and returns a character resource according to the format.

```
string str = cMessageDataSheetMgr.format(new MulId(0,0,1),"Test",n,value);
```

11.34 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.

class CWindowMgr

void **setUIResolution** (float *width*, float *height*)

Param width Sets the screen width of the UI. If the value is 0, the scaling factor is calculated relative to the height.

Param height Sets the screen height of the UI.Only If width = 0, calculates the scaling factor with respect to the height.

Sets the screen height of the UI. Only if width = 0,calculates the scaling factor with respect to the height.

- Case: width != 0

Determines the UI screen size relative to width.

- Case: width = 0 and height != 0

Determines the UI screen size relative to height.

- Case: width = 0 and height = 0(Default)

It conforms to the screen size of the terminal.

bool **initialize** (*CAssetBundle*[] *aAB*)

Expand and initialize the loaded window resource.If CassetBundleMgr is registered, it will automatically download the window resources from the server and pass them to this function.

bool **load** (byte[] *buffer*)

Expands and initializes window resources loaded as byte data.

bool **load** (string *assetname*)

Param string assetname Name of the imported asset

Loads and initializes the window resource with the specified asset name from the resource data.

The actual imported asset is named assetname + ".asset".

WindowScript **create**<WindowScript> (uint *id*, *CWindowBase cParent* = *null*)

The class name of the window to create. The class name of the window to create.

Param uint id Window ID

Param CWindowBase cParet Parent window

Creates a window of the specified class.

WindowScript **find**<WindowScript> (uint *id*)

The class name of the window to create. Window class name

Param uint id Window ID

Finds the specified window. If not yet created, null is returned.

void **bringToTop** (*CWindowBase cWindow*)

Param CWindowBase cWindow target window

rise a priority of target window.

string **getCaption** (uint *mCaptionId*)

Param uint mCaptionId Caption ID

Gets the string from the specified caption ID.

void **play** (uint *mSE*)

Param uint mSE SE's ID

Play the SE

uint **clickSE** { **get; set;** }

Specifies a standard sound to be played when touched.

uint **scrollSE** { **get; set;** }

Specifies a standard sound to be played when scrolling through a list box.

static CWindowMgr **Instance** { **get;** }

Gets an instance of a window manager.

11.35 CWindowBase

11.35.1 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 = NOTITLEBAR|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();
```

- The callback that reach to each window object

class CWindowBase

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.

You need to create new class which is inherited the class that is generated when you compile the script window. And you override callback functions in this new class.

virtual void **onCreate** ()

called immediately after the window has been initialized.

virtual void **onUpdate**()

Called before each control update.

virtual void **onAfterUpdate**()

Called after each control update.

virtual void **onPreRender**(CWindowRenderer *cRenderer*)

Called before each control render.

virtual **onRender**(CWindowRenderer *cRenderer*)

Called after each control render.

virtual bool **onClose**(int *iCloseInfo*)

Called when the window is closed.

- true: close window
- false: Stop closing window

virtual void **onClick**(*CWinCtrlBase* *cCtrl*)

Control is pressed.

virtual void **onHold**(*CWinCtrlBase* *cCtrl*)

Control is hold.

virtual void **onClickEnter**(*CWinCtrlBase* *cCtrl*)

Input of edit box is confirmed.

virtual void **onClick**(*CWinCtrlBase* *cCtrl*, *CRichTextOne* *cText*)

A part of the rich text is pressed

virtual void **onBeginDrag**(*CWinCtrlBase* *cCtrl*, Vector2 *pos*)

The control is dragged. called once first when it is dragged.

virtual void **onDrag**(*CWinCtrlBase* *cCtrl*, Vector2 *pos*)

The control is being dragged. kept calling while dragging.

You can tell which direction a scrollable control has been dragged by onDrag.However, STYLE must have *SCROLL_LOCK* .onDrag can detect the direction of the drag.Note that DRAG does not affect these controls.

virtual bool **onDragRender**(*CWinCtrlBase* *cCtrl*, Transform *transform*)

When a control is being dragged, called the render timing. If not overridden, the control is rendered normally.

The return value has the following meaning:

- true: Renders a duplicate of the control being dragged.
- false: Do not render duplicates of the controls being dragged.

virtual void **onDrop**(*CWinCtrlBase* *cCtrl*, *CWindowBase* *cDragWindow*, *CWinCtrlBase* *cDragCtrl*)

When the control being dragged is released, it is called on the window containing the control that exists under the released location.

virtual void **onDropGround**(*CWinCtrlBase* *cCtrl*)

Called when control being dragged is released in empty space.

It is also called when the window containing the control being dragged has disappeared, just before the disappearance.

virtual void **onOK** (int *msgBoxWinID*)

OK button was pressed in the message window that was created as a child window.

virtual void **onNext** (int *msgBoxWinID*)

Next button was pressed in the message window that was created as a child window.

virtual void **onYes** (int *msgBoxWinID*)

Yes button was pressed in the message window that was created as a child window.

virtual void **onNo** (int *msgBoxWinID*)

No button was pressed in the message window that was created as a child window.

virtual void **onCancel** (int *msgBoxWinID*)

Cancel button was pressed in the message window that was created as a child window.

virtual void **onRecreatedRenderTexture** (uint *mRenderTextureId*)

It is called when a render texture disappears due to some event. However, since the automatic reset is performed by the system, it is not necessary to override the automatic reset.

- Window property/method

I explain the accessible methods and properties of CWindowBase.

virtual bool **onBeginRenderIcon** (*CWinCtrlRenderIcon* *icon*)

Start rendering to the icon. `CWinCtrlRenderIcon` is called only when the window contains it.

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.

virtual bool **onEndRenderIcon** (*CWinCtrlRenderIcon* *icon*)

Called when rendering is finished. `CWinCtrlRenderIcon` is called only when the window contains it.

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.

void **close** (int *iCloseInfo = 0*)

Call when you want to close a window. iCloseInfo is passed as a parameter to onClose.

WinCtrl **find**<WinCtrl> (uint *id*)

Find the control with that ID.

CWinCtrlBase **find** (uint *id*)

Find the control with that ID.

uint **id** { **get; set;** }

Get/Set the window ID.

CWindowBase **child** { **get;** }

return child window.

CWindowBase **parent** { **get;** }

return parent window.

e_LayerId **layer** { **get; set;** }

Get/Set rendering layer.

int **priority** { **get; set;** }

Get/Set drawing priority.

uint **texid** { **get; set;** }

return the default texture id of this window.

float **x** { **get; set;** }

float **y** { **get; set;** }

Get/Set window coordinate

Vector2 **position** { **get; set;** }

Get/Set window coordinate

Transform **transform** { **get;** }

Get transform of this window.

Vector2 **scale** { **get; set;** }

Get/Set x,y scale of this window.

e_Anchor **anchor** { **get; set;** }

Get/Set the display position anchor setting.

float **width** { **get; set;** }

float **height** { **get; set;** }

Get/Set this window size.

Vector2 **size** { **get; set;** }

Get/Set this window size.

float **screenWidth** { **get; set;** }

float **screenHeight** { **get; set;** }

Get/Set this window screen size.

string **caption** { **get; set;** }

Get/Set a title bar's caption of this window.

WinColor **captionColor** { **get; set;** }

Get/Set a title bar's color fo this window.

Vector2 **captionOffset** { **get; set;** }

Get/Set a title bar's caption offset of this window.

WinColor **color** { **get; set;** }

Get/Set this window color. this color is all affect all contorl under the window.

e_WinStyle **style** { **get; set;** }

Get this window style.

e_WinStyle **priorityStyle** { **get; set;** }

Get/Set this window priority style.

bool **hide** { **get; set;** }

- true

  Hide window.

- false

  How to create a window class

bool **disable** { **get; set;** }

- true

  Stop a window from functioning and dim the window.

- false

  Display the window normally.

if stop a window from functioning, it won't respond when you touch it.

bool **isClose** { **get;** }

True if the window is during close animation.

bool **isLoaded** { **get;** }

True if textures are being imported.

bool **isFade** { **get;** }

True if the window is fade-in/fade-out.

- 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*）

Called when a fade begins.The fade-in/fade-out can be determined by looking at e _ FadeState.Perform the initialization required for fade processing.

- e_FadeState.Open

  Fade-In

- e_FadeState.Close

Fade-Out

protected virtual bool **doFade** (e_FadeState *eState*)

Called during fade.

- true

  Fading

- false

  Fade End

When the fade is finished, you need to return false.

Also, when e_FadeState.None is passed, it should return false.

- e_FadeState.None

  No Fading.

- e_FadeState.Open

  Fading In

- e_FadeState.Close

  Fading Out

11.36 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.

11.36.1 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);
```

11.36.2 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);
```

class CWinCtrlBase

For the contents

CWinContents **getContents** ()

return CWinContents Gets the *contents* defined by the *CONTENTS* properties.

For the contents list

int **count** { **get;** }

Get the number of content lists.

void **resize** (int *num*)

Param int num Contents list

Set the number of content lists.

As the number of content lists increases, duplicate the content as needed as a template.

It does not work with elements in content lists that already exist.

CWinContents **insert** (int *pos*)

Param int pos Where to create and insert content into the content list.

Return CWinContents :type:Content<CWinContents>‘ in the content list

Generate and insert content to a specific location in the content list.

CWinContents **getContentsFromIndex** (int *index*)

Param int index Index of the content you want to get in the content list.

Return CWinContents :type:Content<CWinContents>‘ in the content list

Get from a content list by index.

int **getContentsIndex** (*CWinCtrlBase ctrl*)

Returns the index of the content list that contains the specified control.If it is not in any content list element, -1 is returned.

int **getContentsIndex** (*CWinContents contents*)

Returns the index of the content list that contains the specified content.If it is not in any content list element, -1 is returned.

Others

string **ToString** ()

Generate a string by control type and ID.

uint **id** { **get; set;** }

Get/Set the ID of the control.

e_WinCtrlKind **kind** { **get;** }

Return the type of control.

int **priority** { **get; set;** }

Get/Set the priority of the control.

Vector3 **position** { **get; set;** }

Set the coordinates of a control.

Vector2 **absPosition** { **get; set;** }

Get Control Coordinates From Window Base

Vector2 **screenPosition** { **get; set;** }

Get Control Coordinates From Screen Reference

float **width** { **get; set;** }

float **height** { **get; set;** }

Get/Set the size of the control.

Vector2 **size** { **get; set;** }

Get/Set the size of the control.

Vector2 **contentsSize** { **get; set;** }

Get/Set the size of the content.

WinColor **getColor** (int *index*)

index = 0 . . . 7. Get the color. A control can have up to 8 texture parts, so you can have up to 8 colors.

void **setColor** (int *index*, WinColor *color*)

index = 0 . . . 7. Set the color. A control can have up to 8 texture parts, so you can have up to 8 colors.

WinColor **color** { **get; set;** }

Get/set the color of texture part number 0.

WinColor **color1** { **get; set;** }

WinColor **color2** { **get; set;** }

WinColor **color3** { **get; set;** }

WinColor **color4** { **get; set;** }

WinColor **color5** { **get; set;** }

WinColor **color6** { **get; set;** }

WinColor **color7** { **get; set;** }

Get/set the color of texture part number 1 (... 7).

void **setTextureId** (int *index*, uint *texid*)

index = 0 ... 7. Set a specific texture ID.Since a control can have up to 8 texture parts, it can also have 8 corresponding texture IDs.

WinColor **getTextureId** (int *index*)

index = 0 ... 7. Set a specific texture ID.Since a control can have up to 8 texture parts, it can also have 8 corresponding texture IDs.

uint **texId** { **get; set;** }

uint **texId1** { **get; set;** }

Get/Set the Texture ID.

void **setPartId** (int *index*, uint *partId*)

index = 0 ... 7. Set a specific texture part ID.Since a control can have up to 8 texture parts, it can also have 8 corresponding texture part IDs.

uint **getPartId** (int *index*)

index = 0 ... 7. Set a specific texture part ID.Since a control can have up to 8 texture parts, it can also have 8 corresponding texture part IDs.

void **setTextureSize** (int *index*, Vector2 *size*)

index = 0 ... 7. Set a specific texture part size.Since a control can have up to 8 texture parts, it can also have 8 corresponding texture sizes.In particular, when index = 0, it is equivalent to accessing width, height.

uint **parts** { **get; set;** }

uint **partId1** { **get; set;** }

uint **partId2** { **get; set;** }

uint **partId3** { **get; set;** }

uint **partId4** { **get; set;** }

uint **partId5** { **get; set;** }

uint **partId6** { **get; set;** }

uint **partId7** { **get; set;** }

Get/Set the Texture part ID.

void **setTextureOffset** (int *index*, Vector2 *offset*)

index = 0 ... 7. Set a specific texture part offset.Since a control can have up to 8 texture parts, it can also have 8 corresponding texture part offsets.

Vector2 **getTextureOffset** (int *index*)

index = 0 ... 7. Set a specific texture part offset.Since a control can have up to 8 texture parts, it can also have 8 corresponding texture part offsets.

e_Anchor **anchor** { **get; set;** }

Get/Set the anchor for determining the origin position.

e_Anchor **baseAnchor** { **get; set;** }

Get/Set the base anchor that determines where the control's origin will be.

e_Anchor **textAnchor** { **get; set;** }

Get/set the text anchor that determines where the caption's origin will be.

uint **fontKind** { **get; set;** }

Get/Set the font type.

string **caption** { **get; set;** }

Get/Set the caption.

uint **captionId** { **get;** }

Get/Set the Caption ID. However, this does not affect caption.

WinColor **captionColor** { **get; set;** }

Get/Set the caption color.

Vector2 **captionOffset** { **get; set;** }

Set the display offset of the caption.

float **lineSpace** { **get; set;** }

Set the spacing between content. In rich text, for example, this is the value between lines.

uint **SEId** { **get; set;** }

Get/Set the ID of the SE assigned to the control.

CWinContents **parent** { **get; set;** }

Gets the content that contains the control.

CWinCtrlBase[] **groups** { **get;** }

Gets the GROUP that the control has.

CWindowBase **window** { **get;** }

Return the window that contains the control.

e_WinCtrlStyle **style** { **get;** }

Gets the style flag for a control.

bool **hide** { **get; set;** }

true: Turns off the display of the control.

false: Show controls.

bool **disable** { **get; set;** }

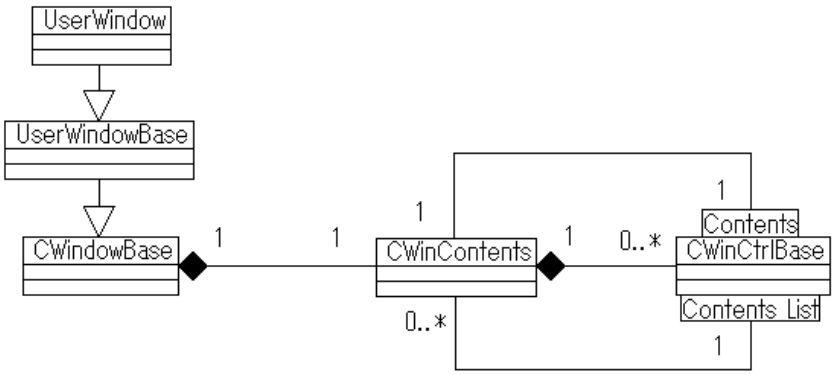
- true: makes the control inoperable and darkens.
- false: The control is normally visible.

bool **nohit** { **get; set;** }

- true: makes the control inoperable(Not as dark as disable).
- false: The control is normally visible.

11.37 CWinContents

class 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.

Functions / Properties

WinCtrl **find**<WinCtrl>（uint *ctrlId*）

> **Param uint ctrlId** Control ID

Locate the control with the specified ID. If it doesn't exist in the content, it recursively looks in the content of the control.However, the content list is not included in the search. This ensures that the control is unique.

int **getIndex**（uint *ctrlId*）

> **Param uint ctrlId** Control ID

Locates the position of the specified control ID in the list. If not found, -1 is returned. Unlike find, it does not search recursively.

The returned index value can be used with this [N] for fast access.

CWinCtrlBase **this**[, int *N*] { **get;** }
You can use the index value obtained by " getIndex " .

int **count** { **get;** }
Returns the number of controls in the content.

ClipRect.State **clipState** { **get;** }
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;** }
Gets the width of the content. Automatically calculated from the size of the control.

float **height** { **get;** }
Gets the height of the content. Automatically calculated from the size of the control.

Vector2 **size** { **get;** }
Gets the size of the content. Automatically calculated from the size of the control.

Vector3 **position** { **get;** }
Display position of the content

CWinCtrlBase **parent** { **get;** }
Gets the control that is the parent of the content.

if the parent of this contents is *CWindowBase* ,it returns a null.

11.38 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* .

Callbacks arriving at `CWindowBase`

This control's style is set NOHIT flag as default. If you want to receive onDrag callback,you need to set HIT flag to STYLE property.

```
void onClick(CWinCtrlBase cCtrl)

void onHold(CWinCtrlBase cCtrl)

void onBeginDrag(CWinCtrlBase cCtrl,Vector2 pos)

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2
dragVelocity)

bool onDragRender(CWinCtrlBase cCtrl,Transform transform)

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,
CWinCtrlBase cDragCtrl)

void onDropGround(CWinCtrlBase cCtrl)
```

Functions / Properties

None

Example of use

```
CWinCtrlText   ctrlName = find<CWinCtrlText>(TEXT_Name);
ctrlName.caption = "Mike";
```

11.39 CWinCtrlRichText

Script: *RICHTEXT*

It is a control for displaying rich text.

It is possible to change the text color , insert some texture part, and change the font type

Callbacks arriving at **:class:‘CWindowBase<CWindowBase>‘**

void onClick(CWinCtrlRichText cCtrl,CRichTextOne cText)

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)

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,CWinCtrlBase cDragCtrl)

class CRichTextOne

Functions / Properties

This is the argument passed to the click event. By examining the member, it is possible to determine where clicked .

CRichTextOne.e_Cmd **cmd** { **get;** }

e_Cmd.String

text: This text part does not trigger a click event.

e_Cmd.Window

Window: \w *command* part is clicked.

The following properties contain valid values.

userId

windowId

e_Cmd.Texture

Texture: \t *command* part is clicked.

The following properties contain valid values.

userId

texId

partId

uint **userId** { **get;** }

Contains the user ID set in the caption. If the user ID is omitted, 0 is used.

uint **windowId** { **get;** }

Gets the window ID specified by \w *command* .

uint **texId** { **get;** }

Gets the texture ID specified by \t *command* .

uint **partId** { **get;** }

Gets the texture part ID specified by \t *command* .

11.40 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.

Callbacks arriving at **:class:'CWindowBase<CWindowBase>'**

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2 dragVelocity)

Called when you scroll . Even if you *lock* the scroll , this callback is called .

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,CWinCtrlBase cDragCtrl)

class CWinCtrlLog

Functions / Properties

void **resize** (int *count*)

Param int count Log's size

Listbox returns the number of contents, while Log is the maximum number of logs that can be kept.

int **count** { **get;** }

For Listbox it is the number of content in the content list, but for Log it is the number of logs it have.

It is never greater than the value set in resize.

void **clearLog** ()

Clears all currently held logs and empties them.

CWinCtrlLogText **add** (string *text*, WinColor *color*)

Param string text Text to add

Param WinColor color Text color

Return CWinCtrlLogText Log text control with string

To add a log, you first add content internally (or reuse it).

Finds the *CWinCtrlLogText* in its content and sets the text and color specified for the control's caption.

You can get the added content from the returned CWinCtrlLogText.

```
CWinCtrlLogText ltChat = lgChat.add(strSentence,WinColor.white);
//Access to content that contains the added LogText.
CWinContents contents = ltChat.parent;
```

Vector2 **viewsize** { **get; set;** }

It is the size of the list box.It is equivalent to the size property.

Vector2 **screensize** { **get; set;** }

It is the size of the list box virtual screen.This value is the sum of all heights in the content list.

Vector2 **offset** { **get; set;** }

This value represents where the list box is pointing in the virtual screen. Set a value to move there instantly.If you want to move the animation smoothly,you can use *setSmoothOffset* .

bool **isSwipe** { **get;** }
This value determines whether the user is swiping.

void **setSmoothOffset** (Vector2 *offset*, float *spd*)

Param Vector2 offset Final offset you want to set

Param float spd Veloccity

Vector2 **smoothOffset** { **get; set;** }
It behaves the same as calling *setSmoothOffset* at the speed of 0.25 .

bool **isSmoothScrolled** { **get;** }
You can check this value with to see if you are during setSmoothOffset scrolling animation.

Vector2 **getContentsOffset** (int *index*, e_Anchor *eAnchor*)

Param int index Index of the content offset you want to get.

Param e_Anchor eAnchor Specify where you want the content to be placed with the anchor

Specify where you want the content to be placed with the anchor.

If the contents of the list box are arranged in a horizontal 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

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;
```

How to clear log.

```
//log clear
lgChat.clearLog();
```

11.41 CWinCtrlLogText

Script: *LOGTEXT*

This is a control for having the sentence of the chat log. This behaviour is same as *TEXT* .

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

Callbacks arriving at **:class:'CWindowBase<CWindowBase>'**

This control's style is set NOHIT flag as default. If you want to receive onDrag callback,you need to set HIT flag to STYLE property.

```
void onClick(CWinCtrlBase cCtrl)

void onHold(CWinCtrlBase cCtrl)

void onBeginDrag(CWinCtrlBase cCtrl,Vector2 pos)

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2
dragVelocity)

bool onDragRender(CWinCtrlBase cCtrl,Transform transform)

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,
CWinCtrlBase cDragCtrl)

void onDropGround(CWinCtrlBase cCtrl)
```

Functions / Properties

None

Example of use

reference *CWinCtrlLog's example*

11.42 CWinCtrlEditbox

Script: *EDITBOX*

It is a text editable control.

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

Callbacks arriving at **:class:'CWindowBase<CWindowBase>'**

```
void onClick(CWinCtrlBase cCtrl)

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2 dragVelocity)

void onClickEnter(CWinCtrlBase cCtrl)
```

It is called when a input string has been determined . The input string is stored in the caption.

```
void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,CWinCtrlBase
cDragCtrl)
```

This callback sent to the window of cCtrl.

Example of use

```
// get a control.
CWinCtrlEditbox        ebSentence = find<CWinCtrlEditbox>(EDITBOX_Sentence);

// get entered text.
string strSentence = ebSentence.caption;


// Called when input is complete..
override protected void onClickEnter(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
      case EDITBOX_Sentence:
        break;
    }
}
```

11.43 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.

Callbacks arriving at `CWindowBase`

```
void onClick(CWinCtrlBase cCtrl)
```

```
void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2 dragVelocity)
```

```
void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,CWinCtrlBase
cDragCtrl)
```

Functions / Properties

None

Example of use

11.44 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.

Callbacks arriving at **:class:'CWindowBase<CWindowBase>'**

```
void onClick(CWinCtrlBase cCtrl)
```

```
void onHold(CWinCtrlBase cCtrl)
```

```
void onBeginDrag(CWinCtrlBase cCtrl,Vector2 pos)
```

```
void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2 dragVelocity)
```

```
bool onDragRender(CWinCtrlBase cCtrl,Transform transform)
```

```
void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,CWinCtrlBase
cDragCtrl)

void onDropGround(CWinCtrlBase cCtrl)
```

Example of use

```
// get control
CWinCtrlButton ctrlStart = find<CWinCtrlButton>(BUTTON_Start);

// click call back
override protected void onClick(CWinCtrlBase cCtrl) {
 switch (cCtrl.id) {
  case BUTTON_Start:
    break;
 }
// hold callback.
override protected void onHold(CWinCtrlBase cCtrl) {
 switch (cCtrl.id) {
  case BUTTON_Start:
    break;
 }
```

11.45 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.

Callbacks arriving at **:class:'CWindowBase<CWindowBase>'**

```
void onClick(CWinCtrlBase cCtrl)

void onHold(CWinCtrlBase cCtrl)

void onBeginDrag(CWinCtrlBase cCtrl,Vector2 pos)

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2 dragVelocity)

bool onDragRender(CWinCtrlBase cCtrl,Transform transform)

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,CWinCtrlBase
cDragCtrl)

void onDropGround(CWinCtrlBase cCtrl)
```

class CWinCtrlRadio

Functions / Properties

bool **check** { **get; set;** }

This value changes the state of the radio button.

- true:

  Turns ON and OFF the grouped radio buttons. Activate any associated content.

- false:

  Turn OFF and hide any associated content.

Example of use

```
// get control
CWinCtrlRadio  rdSelect = find<CWinCtrlRadio>(RADIO_Select);

if (rdSelect.check) {
  //ON
} else {
  //OFF
}

// click callback
override protected void onClick(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
      case RADIO_Select:
        break;
    }
}
// hold callback
override protected void onHold(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
      case RADIO_Select:
        break;
    }
}
```

11.46 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.

Callbacks arriving at **:class:'CWindowBase<CWindowBase>'**

void onClick(CWinCtrlBase cCtrl)

void onHold(CWinCtrlBase cCtrl)

void onBeginDrag(CWinCtrlBase cCtrl,Vector2 pos)

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2 dragVelocity)

bool onDragRender(CWinCtrlBase cCtrl,Transform transform)

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,CWinCtrlBase cDragCtrl)

void onDropGround(CWinCtrlBase cCtrl)

class CWinCtrlCheckBox

bool **check** { **get; set;** }

Change a check box status.

- true:

Turn ON and active any associated content.

- false:

  Turn OFF and hide any associated content.

Example of use

```
// get control
CWinCtrlCheckbox       cbSelect = find<CWinCtrlCheckbox>(CHECKBOX_Select);

if (cbSelect.check) {
  //on
} else {
  //off
}
// Get BUTTON(A) from within content
// Content in the container can be accessed by find as follows
CWinCtrlButton btnA = find<CWinCtrlButton>(BUTTON_A);

// click callback
override protected void onClick(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
      case CHECKBOX_Select:
        break;
    }
}
 // hold callback
override protected void onHold(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
      case CHECKBOX_Select:
        break;
    }
}
```

11.47 CWinCtrlTexture

Script: *TEXTURE*

This is a control for displaying a texture part .

It is possible to display eight texture parts.

Callbacks arriving at `CWindowBase`

This control's style is set NOHIT flag as default. If you want to receive onDrag callback,you need to set HIT flag to STYLE property.

```
void onClick(CWinCtrlBase cCtrl)

void onHold(CWinCtrlBase cCtrl)

void onBeginDrag(CWinCtrlBase cCtrl,Vector2 pos)

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2
dragVelocity)

bool onDragRender(CWinCtrlBase cCtrl,Transform transform)
```

```
void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,
CWinCtrlBase cDragCtrl)

void onDropGround(CWinCtrlBase cCtrl)
```

Functions / Properties

None

Example of use

```
// get control
CWinCtrlTexture        texAttribute = find<CWinCtrlTexture>(TEXTURE_Attribute);

// click callback
override protected void onClick(CWinCtrlBase cCtrl) {
  switch (cCtrl.id) {
    case TEXTURE_Attribute:
      break;
   }
}
// hold callback
override protected void onHold(CWinCtrlBase cCtrl) {
   switch (cCtrl.id) {
     case TEXTURE_Attribute:
      break;
    }
}
```

11.48 CWinCtrlLine

Script: *LINE*

This is the control to draw a line segment.

Callbacks arriving at **:class:'CWindowBase<CWindowBase>'**

This control's style is set NOHIT flag as default. If you want to receive onDrag callback,you need to set HIT flag to STYLE property.

```
void onClick(CWinCtrlBase cCtrl)

void onHold(CWinCtrlBase cCtrl)

void onBeginDrag(CWinCtrlBase cCtrl,Vector2 pos)

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2
dragVelocity)

bool onDragRender(CWinCtrlBase cCtrl,Transform transform)

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,
CWinCtrlBase cDragCtrl)

void onDropGround(CWinCtrlBase cCtrl)
```

Example of use

```
// get control
CWinCtrlLine   lnTest = find<CWinCtrlLine>(LINE_TEST);
```

11.49 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.

Callbacks arriving at **:class:'CWindowBase<CWindowBase>'**

void onClick(CWinCtrlBase cCtrl)

void onHold(CWinCtrlBase cCtrl)

void onBeginDrag(CWinCtrlBase cCtrl,Vector2 pos)

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2 dragVelocity)

bool onDragRender(CWinCtrlBase cCtrl,Transform transform)

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,CWinCtrlBase cDragCtrl)

void onDropGround(CWinCtrlBase cCtrl)

class CWinCtrlRender

Functions / Properties

Camera **camera** { **get;** }

you can get the camera to which the rendering texture is assigned. Access when you want to customize the camera position and angle.

Note: Be careful not to create a conflict between the layer settings of the drawing object and the camera.Also, if you can have multiple RENDER controls on the screen at the same time, you must assign them with care to avoid duplicate layer settings in each RENDER.

Example of use

```
// get control
public override void onCreate() {
  CWinCtrlRender       rdAvatar = find<CWinCtrlRender>(RENDER_AVATAR);

  // get camera
  Camera       camera = rdAvatar.camera;

  // set display layer.
  camera.cullingMask = (int) e_Layer.OwnPlayer;

  // set camera position and angle.
  camera.transform.position = new Vector3(0f,0.85f,0f);
  camera.transform.rotation = Quaternion.Euler(new Vector3(0f,180f,0f));
```

(continues on next page)

(continued from previous page)

```
  GameObject goAvatar = LoadAvatar();
  // Sets the layer of game objects to render (Make Same as Camera).
  Utility.setLayer(goAvatar,e_LayerId.OwnPlayer);
}

// click callback
override protected void onClick(CWinCtrlBase cCtrl) {
  switch (cCtrl.id) {
    case RENDER_AVATAR:
      break;
   }
}
// hold callback
override protected void onHold(CWinCtrlBase cCtrl) {
   switch (cCtrl.id) {
     case RENDER_AVATAR:
      break;
    }
}
```

11.50 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.

Callbacks arriving at **:class:‘CWindowBase<CWindowBase>‘**

void onClick(CWinCtrlBase cCtrl)

void onHold(CWinCtrlBase cCtrl)

void onBeginDrag(CWinCtrlBase cCtrl,Vector2 pos)

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2 dragVelocity)

bool onDragRender(CWinCtrlBase cCtrl,Transform transform)

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,CWinCtrlBase cDragCtrl)

void onDropGround(CWinCtrlBase cCtrl)

Example of use

```
// get cotrol
CWinCtrlIcon   icnItem = find(ICON_Item) as CWinCtrlIcon;

// click callback
override protected void onClick(CWinCtrlBase cCtrl) {
  switch (cCtrl.id) {
    case ICON_Item:
      break;
```

(continues on next page)

(continued from previous page)

```
    }
}
// hold callback
override protected void onHold(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
      case ICON_Item:
        break;
    }
}
```

11.51 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.

Callbacks arriving at **:class:'CWindowBase<CWindowBase>'**

void onClick(CWinCtrlBase cCtrl)

void onHold(CWinCtrlBase cCtrl)

void onBeginDrag(CWinCtrlBase cCtrl,Vector2 pos)

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2 dragVelocity)

bool onDragRender(CWinCtrlBase cCtrl,Transform transform)

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,CWinCtrlBase cDragCtrl)

void onDropGround(CWinCtrlBase cCtrl)

class CWinCtrlRecastIcon

Functions / Properties

CInput::e_State **state** { **get;** }
Gets the state that the icon is depressed.

void **setRecastTime** (int *index*, float *tmRecast*)

Param int index Texture part number for the recast animation

Param float tmRecast Animation Time [0 . . . 1]

Set a recast time.

public float **getRecastTime** (int *index*)

Param int index Texture part number for the recast animation

Return float remaining animation time

Get a recast time.

Example of use

```
// get control
CWinCtrlRecastIcon     rcItem = find(RECASTICON_Item) as CWinCtrlRecastIcon;

// Recast Display (Display half of the texture of TEX_ID3)
rcItem.setRecastTime(3,0.5f);

// click callback
override protected void onClick(CWinCtrlBase cCtrl) {
  switch (cCtrl.id) {
    case RECASTICON_Item:
      break;
   }
}
// hold callback
override protected void onHold(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
      case ICON_Item:
        break;
    }
}
```

11.52 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.

Callbacks arriving at **:class:‘CWindowBase<CWindowBase>‘**

void onClick(CWinCtrlBase cCtrl)

void onHold(CWinCtrlBase cCtrl)

void onBeginDrag(CWinCtrlBase cCtrl,Vector2 pos)

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2 dragVelocity)

bool onDragRender(CWinCtrlBase cCtrl,Transform transform)

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,CWinCtrlBase cDragCtrl)

void onDropGround(CWinCtrlBase cCtrl)

bool onBeginRenderIcon(CWinCtrlRenderIcon icon)

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)

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.

class CWinCtrlRenderIcon

Functions / Properties

ulong **regionId** { **get; set;** }

Assign a unique ID other than 0. Setting it to 0 frees the reserved render texture.Also, if you assign the same value as other controls, they share a render texture.

Camera **camera** { **get;** }

you can get the camera to which the rendering texture is assigned. Access when you want to customize the camera position and angle.

WinColor **color1** { **get; set;** }

It is assigned as the background color for rendering.

Default valus is (0,0,0,0).

bool **needToRender** { **get; set;** }

Set to true when you want to restart rendering.

If you return a true, rendering of the next frame is continued(onBeginRenderIcon is called at the next frame.).

CInput::e_State **state** { **get;** }

Gets the state that the icon is depressed.

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?
- Rendering targets are within the camera?

  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 = 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_lstGameObject[idx];
    if (go == null) {
       return false;
    }
    go.SetActive(false);
    return false;
}

// click callback
override public void onClick(CWinCtrlBase cCtrl) {
  switch (cCtrl.id) {
    case RENDERICON_Avatar:
      break;
   }
}
// hold callback
override public void onHold(CWinCtrlBase cCtrl) {
   switch (cCtrl.id) {
     case RENDERICON_Avatar:
      break;
    }
}
```

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.

```
// If gameobjectAvatar is not multiple hierarchies, the following is sufficient.
gameobjectAvatar.layer = (int) e_LayerId.RenderIcon;
// If gameobjectAvatar is made up of multiple hierarchies, it must be recursively set␣
↪using the following function:
Utility.setLayer(gameObjectAvatar,e_LayerId.RenderIcon);
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.

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.

Timing onBeginRender is called .

- You set a non-zero value to regionId.

  Immediately after setting `regionId` to a unique non-zero value. `onBeginRenderIcon` is called.It will continue to be called until `onBeginRenderIcon` returns true. If it returns false, `onEndRenderIcon` will not be called.Using this mechanism, during an asynchronous import of a model it is possible to return false and render the model to texture when it is ready to be rendered (Don't forget to set the layer).

- 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.

11.53 CWinCtrlMeter

Script: *METER*

This is the control to display a meter.

You can use this as a progress bar.

As with *TEXTURE* , it is possible to display up to 8 images on top of each other.

You can stretch any texture parts.

If the length is set to 1, it display same look as *TEXTURE* .

Callbacks arriving at **:class:'CWindowBase<CWindowBase>'**

METER control's style is set NOHIT flag as default. If you want to receive onDrag callback,you need to set HIT flag to STYLE property.

```
void onClick(CWinCtrlBase cCtrl)

void onHold(CWinCtrlBase cCtrl)

void onBeginDrag(CWinCtrlBase cCtrl,Vector2 pos)

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2
dragVelocity)

bool onDragRender(CWinCtrlBase cCtrl,Transform transform)

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,
CWinCtrlBase cDragCtrl)

void onDropGround(CWinCtrlBase cCtrl)
```

class CWinCtrlMeter

Functions / Properties

float **meter** (int *index*)

Param int index Texture part number

Return float meter's length

Gets the percentage of the original size that the specified a texture part.

It returns a value in the range of 0 . . . 1.

By default, this is set to 1 (status of the meter at the max).

void **setMeter** (int *index*, float *value*, float *speed = 0.2f*)

Param int index Texture part number

Param float value meter's length(0 . . . 1)

Param float speed = 0.2f animation speed

Changes the size of the specified texture part with animation.

Example of use

```
// get control
CWinCtrlMeter  mtrLoading = find(METER_Loading) as CWinCtrlMeter;

// Get Meter Position (0 ... 1.)
float pos = mtrLoading.meter(3);

// Set 3rd texture part stretch (It can be specified as 0 ... 1.)
mtrLoading.setMeter(3,0.5f);
```

11.54 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.

Callbacks arriving at **:class:'CWindowBase<CWindowBase>'**

None

class CWinCtrlScrollbar

Functions / Properties

int **index** { **get; set;** }

Specifies the texture index that acts as a scrollbar.

Textures other than those specified by this value are displayed without length or position changes.

float **fadeSpeed** { **get; set;** }

Specifies the speed at which the scroll bar disappears when it is *no longer displayed* .

Example of use

```
CWinCtrlScrollbar     sbListbox = find<CWinCtrlScrollbar>(SCROLLBAR_Listbox);
```

11.55 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 .

Callbacks arriving at **:class:‘CWindowBase<CWindowBase>‘**

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2 dragVelocity)

Called when you scroll . Even if you *lock* the scroll , this callback is called .

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,CWinCtrlBase cDragCtrl)

class CWinCtrlListBox

Functions / Properties

Vector2 **viewsize** { **get; set;** }

It is the size of the list box.It is equivalent to the size property.

Vector2 **screensize** { **get; set;** }

It is the size of the list box virtual screen.This value is the sum of all heights in the content list.

Vector2 **offset** { **get; set;** }

This value represents where the list box is pointing in the virtual screen. Set a value to move there instantly.If you want to move the animation smoothly,you can use *setSmoothOffset* .

bool **isSwipe** { **get;** }

This value determines whether the user is swiping.

void **setSmoothOffset** (Vector2 *offset*, float *spd*)

Param Vector2 offset Final offset you want to set

Param float spd Veloccity

Vector2 **smoothOffset** { **get; set;** }

Veloccity

bool **isSmoothScrolled** { **get;** }

You can check this value with to see if you are during setSmoothOffset scrolling animation.

Vector2 **getContentsOffset** (int *index*, e_Anchor *eAnchor*)

Param int index Index of the content offset you want to get.

Param e_Anchor eAnchor Specify where you want the content to be placed with the anchor

Specify where you want the content to be placed with the anchor.

If the contents of the list box are arranged in a horizontal 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

Example of use

```
override public void onUpdate() {
   // get control
   CWinCtrlListbox     lbFriends = find<CWinCtrlListbox>(LISTBOX_Friends);

   // set list box contents num.
   lbFriends.resize(10);

   // Refresh the status in a list box (In this case, 10 lines will be updated.)
   for (int i = 0;i < lbFriends.Count;++i) {
       // Get a single line of list box content.
       CWinContents contents = lbFriends.getContentsFromIndex(i);

       //Get BUTTON(Name)
       CWinCtrlButton btnName = contents.find<CWinCtrlButton>(BUTTON_Name);

       //Get ICON(Icon)
       CWinCtrlIcon icnName = contents.find<CWinCtrlIcon>(ICON_Icon);
   }
}
```

11.56 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.

Callbacks arriving at **:class:'CWindowBase<CWindowBase>'**

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2 dragVelocity)

Called when you scroll . Even if you *lock* the scroll , this callback is called .

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,CWinCtrlBase cDragCtrl)

class CWinCtrlListboxEx

Functions / Properties

Vector2 **viewsize** { **get; set;** }

It is the size of the list box.It is equivalent to the size property.

Vector2 **screensize** { **get; set;** }

It is the size of the list box virtual screen.This value is the sum of all heights in the content list.

Vector2 **offset** { **get; set;** }

This value represents where the list box is pointing in the virtual screen. Set a value to move there instantly.If you want to move the animation smoothly,you can use `setSmoothOffset` .

bool **isSwipe** { **get;** }

This value determines whether the user is swiping.

void **setSmoothOffset** (Vector2 *offset*, float *spd*)

Param Vector2 offset Final offset you want to set

Param float spd Veloccity

Vector2 **smoothOffset** { **get; set;** }

It behaves the same as when you call `setSmoothOffset` at a speed of 0.25 .

bool **isSmoothScrolled** { **get;** }

You can check this value with to see if you are during setSmoothOffset scrolling animation.

Vector2 **getContentsOffset** (int *index*, e_Anchor *eAnchor*)

Param int index Index of the content offset you want to get.

Param e_Anchor eAnchor Specify where you want the content to be placed with the anchor

Specify where you want the content to be placed with the anchor.

If the contents of the list box are arranged in a horizontal 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

Example of use

```
override public void onUpdate() {
   // get control
   CWinCtrlListboxEx   lbPartys = find<CWinCtrlListboxEx>(LISTBOXEX_Partys);

   // set listbox contents num
   lbPartys.resize(10);

   // Refresh the status in a list box (In this case, 10 lines will be updated.)
   for (int i = 0;i < lbFriends.Count;++i) {
       // Get a single line of list box content.
       CWinContents contents = lbFriends.getContentsFromIndex(i);

       //Get BUTTON(Name)
       CWinCtrlButton btnName = contents.find<CWinCtrlButton>(BUTTON_Name);

       //Get ICON(Icon)
       CWinCtrlIcon icnName = contents.find<CWinCtrlIcon>(ICON_Icon);
   }
}
```

11.57 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.

Callbacks arriving at **:class:'CWindowBase<CWindowBase>'**

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2 dragVelocity)

Called when you scroll . Even if you lock the scroll , this callback is called .

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,CWinCtrlBase cDragCtrl)

Example of use

```
// get control
CWinCtrlContainer      ctnMap = find<CWinCtrlContainer>(CONTAINER_Map);
// Get TEXTURE(Map) from within content
// Content in the container can be accessed by find as follows
CWinCtrlTexture txMap = find<CWinCtrlTexture>(TEXTURE_Map);
```

11.58 CWinCtrlFrame

Script: *FRAME*

This is the control to display the frame.

Unlike bar, it does not have a caption.

Callbacks arriving at **:class:'CWindowBase<CWindowBase>'**

void onClick(CWinCtrlBase cCtrl)

void onHold(CWinCtrlBase cCtrl)

void onBeginDrag(CWinCtrlBase cCtrl,Vector2 pos)

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2 dragVelocity)

bool onDragRender(CWinCtrlBase cCtrl,Transform transform)

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,CWinCtrlBase cDragCtrl)

void onDropGround(CWinCtrlBase cCtrl)

Example of use

```
// get control
CWinCtrlFrame  ctrlStart = find(FRAME_BG) as CWinCtrlFrame;

// click callback
override protected void onClick(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
      case FRAME_BG:
        break;
     }
}
```

11.59 CWinCtrlLabel

Script: *LABEL*

This is the control to display the label.Unlike a *BAR* , it does not receive an onClick event.

Callbacks arriving at `CWindowBase`

This control's style is set NOHIT flag as default. If you want to receive onDrag callback,you need to set HIT flag to STYLE property.

```
void onClick(CWinCtrlBase cCtrl)

void onHold(CWinCtrlBase cCtrl)

void onBeginDrag(CWinCtrlBase cCtrl,Vector2 pos)

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2
dragVelocity)

bool onDragRender(CWinCtrlBase cCtrl,Transform transform)

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,
CWinCtrlBase cDragCtrl)

void onDropGround(CWinCtrlBase cCtrl)
```

Example of use

```
// get control
CWinCtrlLabel  ctrlStart = find(LABEL_Title) as CWinCtrlLabel;
```

11.60 CWinCtrlBar

Script: *BAR*

The control to display a bar .

It can have a caption.

Callbacks arriving at **:class:'CWindowBase<CWindowBase>'**

```
void onClick(CWinCtrlBase cCtrl)

void onHold(CWinCtrlBase cCtrl)

void onBeginDrag(CWinCtrlBase cCtrl,Vector2 pos)

void onDrag(CWinCtrlBase cCtrl,Vector2 pos,Vector2 dragVelocity)

bool onDragRender(CWinCtrlBase cCtrl,Transform transform)

void onDrop(CWinCtrlBase cCtrl,CWindowBase cDragWindow,CWinCtrlBase
cDragCtrl)

void onDropGround(CWinCtrlBase cCtrl)
```

Example of use

```
// get control
CWinCtrlBar    ctrlStart = find<CWinCtrlBar>(BAR_Title);

// click callback
override public void onClick(CWinCtrlBase cCtrl) {
    switch (cCtrl.id) {
      case BAR_Title:
        break;
```

(continues on next page)

(continued from previous page)

```
        }
}
```

CHAPTER

TWELVE

APPENDIXES

12.1 How to change the default value

"KsSoft/Plugins/KsSoftConfig.cs" is set various default values.

The default value can be changed in KsSoftConfig.initialize.

12.1.1 Asset bundle

bool **m_UseStreaming** = true
: you can select using assetbundles or resources. Depending on this value, each export changes.Please refer to here

  - If false

    Asset bundle: export path

    KsSoftConfig.m_assetbundlePath value

    Download assets from.

    KsSoftConfig.m_httpserver value

  - If true

    Asset bundle: export path

    KsSoftConfig.m_assetbundlePath value

    Download assets from.

    Where "Assets/StreamingAssets/" is indicated by "*KsSoftConfig.m_assetbundlePath* " if "StreamingAsset" does not exist.

string **m_assetbundlePath** = "assetbundles/"
: Change the output folder of the asset bundle.

string **m_assetbundleExt** = ".unity3d"
: Extension of the asset bundle file.

string **m_resourcePath** = "Assets/Resources/"
: 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.

12.1.2 Window resource

string **m_windowResourcePath** = "Assets/WindowResource/"
Specifies the folder where the 'wra' file is stored.

string **m_WindowResourceBinaryPath** = "wrb/"
Specify a folder to store the results of compiling the 'wra' file.

uint **WindowAssetbundleId**
Window asset bundle ID to download by default

uint **m_defaultFontKind**
The default value for the font ID is used when not specified.

12.1.3 Texture Resources

string **m_textureResourcePath** = "Assets/TextureResource/"
Specifies the folder where texture resources are stored.

12.1.4 Sound Effect Resources.

string **m_SEResourcePath** = "Assets/SE/"
Specify the folder where Sound Effect resources are stored.

12.1.5 BGM Resources

string **m_BgmResourcePath** = "Assets/Bgm/"
Specify the folder where BGM resources are stored.

12.1.6 Fonts

string **m_FontResourcePath** = "Assets/FontResource/"
Specify a folder to store font data to be resourced.

string **m_FontPath** = "Assets/Fonts/"
Specify the folder to store the font data to be asset-bundled.

12.1.7 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)},
};
```

12.1.8 Address/IP

string **httpserver** = "http://xxx.xxx.xxx.xxx/"
Set the address when you download the asset bundle via HTTP

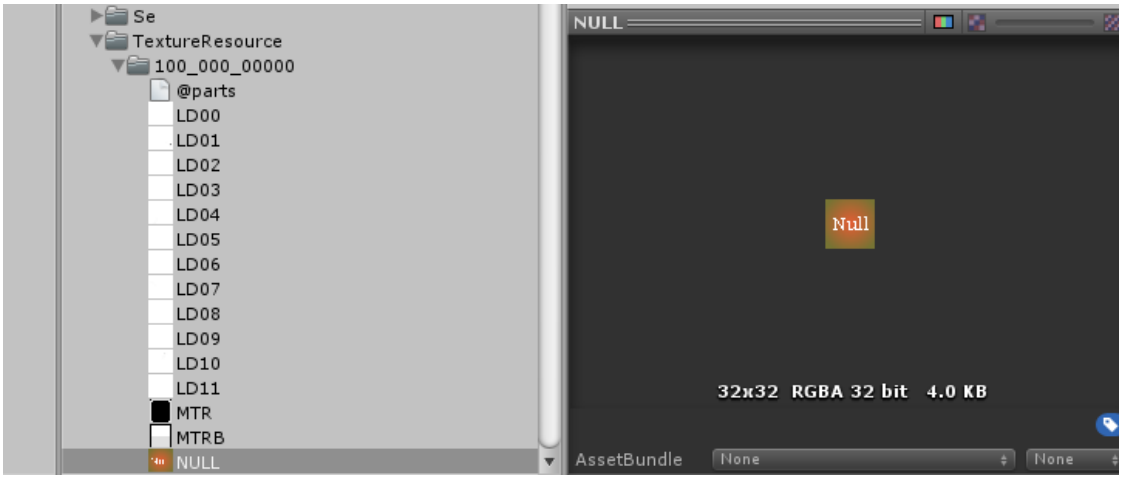
12.2 Texture parts default

Kinds of control	Default value
EDITBOX	0:"TXFD?"(TXFD0,TXFD1) 1:"CURSR"
TEXTBOX	"TXFD"
BUTTON	"BTN0?"(BTN00,BTN01)
RADIO	"BTN0?"(BTN00,BTN01)
CHECKBOX	"BTN0?"(BTN00,BTN01)
TEXTURE	""
LINE	"LINE"
ICON	"ICON"
RECASTICON	"ICON"
METER	0:"MTR" 1:"MTRB"
SCROLLBAR	"SCBRH"
FRAME	"FRAME"
LABEL	"LABEL"
BAR	"BAR"

12.3 "NULL" part: It is used when the texture is not present.

If texture parts do not exist in a texture resource, this window system tries to assign "NULL" texture part.

If it can not find "NULL" texture part, it cancels control rendering.Beforehand if you include "NULL" texture part in the texture resource, you can find out visually that you forget to include some texture parts.



12.4 Callback reaching the CWindowBase

	onClick	onHold	onClickEnter	onBeginDrag	onDrag	onDragRender
TEXT	O			O	O	O
RICHTEXT	O					
LOG					O (Note)	
LOGTEXT	O			O	O	O
EDITBOX	O		O		O (Note)	
TEXTBOX	O				O (Note)	
BUTTON	O	O		O	O	O
RADIO	O	O		O	O	O
CHECKBOX	O	O		O	O	O
TEXTURE	O	O		O	O	O
LINE	O			O	O	O
RENDER	O	O		O	O	O
ICON	O	O		O	O	O
RECASTICON	O	O		O	O	O
RENDERICON	O	O		O	O	O
METER	O			O	O	O
SCROLLBAR						
LISTBOX					O (Note)	
LISTBOXEX					O (Note)	
CONTAINER					O (Note)	
FRAME	O			O	O	O
LABEL	O			O	O	O
BAR	O			O	O	O

Note: You cannot drag rich text and scrollable objects.For scrollable controls, onDrag lets you know in which direction the control was dragged.However, STYLE must have *SCROLL_LOCK* .onDrag can detect the direction you have dragged. Note that DRAG has no effect on these controls.

	onDrop (dst)	onDrop (src)	onDropGround
TEXT	O	O	O
RICHTEXT	O		
LOG	O		
LOGTEXT	O	O	O
EDITBOX	O		
TEXTBOX	O		
BUTTON	O	O	O
RADIO	O	O	O
CHECKBOX	O	O	O
TEXTURE	O	O	O
LINE	O	O	O
RENDER	O	O	O
ICON	O	O	O
RECASTICON	O	O	O
RENDERICON	O	O	O
METER	O	O	O
SCROLLBAR			
LISTBOX	O		
LISTBOXEX	O		
CONTAINER	O		
FRAME	O	O	O
LABEL	O	O	O
BAR	O	O	O

12.5 Non-Dragable Controls

You cannot drag rich text and scrollable objects.For scrollable objects, onDrag lets you know in which direction the control was dragged.

if you set *SCROLL_LOCK* to STYLE property, the window system calls onDrag when you drag these controls.And you can get a dragged direction of these controls.Even if you set DRAG to STYLE property, these controls do not affect anything.

- *RICHTEXT*
- *LOG*
- *EDITBOX*
- *TEXTBOX*
- *SCROLLBAR*
- *LISTBOX*
- *LISTBOXEX*
- *CONTAINER*

12.6 Kinds of valid STYLE flag

	ANCHOR_*	BASE_*	HIDE	DRAG	DISABLE	NOHIT	HIT
TEXT	O	O	O	O	O		O
RICHTEXT	O	O	O	O		O	
LOG	O	O	O	O		O	
LOGTEXT	O	O	O	O	O		O
EDITBOX	O	O	O	O		O	
TEXTBOX	O	O	O	O		O	
BUTTON	O	O	O	O	O	O	
RADIO	O	O	O	O	O	O	
CHECKBOX	O	O	O	O	O	O	
TEXTURE	O	O	O	O	O		O
LINE	O	O	O	O	O		O
RENDER	O	O	O	O	O	O	
ICON	O	O	O	O	O	O	
RECASTICON	O	O	O	O	O	O	
RENDERICON	O	O	O	O	O	O	
METER	O	O	O	O	O		O
SCROLLBAR	O	O	O	O			
LISTBOX	O	O	O	O		O	
LISTBOXEX	O	O	O	O		O	
CONTAINER	O	O	O	O		O	
FRAME	O	O	O	O	O	O	
LABEL	O	O	O	O	O		O
BAR	O	O	O	O	O	O	

	TEXT_LEFT	TEXT_RIGHT	TEXT_CENTER
TEXT			
RICHTEXT	O	O	O
LOG			
LOGTEXT			
EDITBOX			
TEXTBOX			
BUTTON	O	O	O
RADIO	O	O	O
CHECKBOX	O	O	O
TEXTURE			
LINE			
RENDER			
ICON	O	O	O
RECASTICON	O	O	O
RENDERICON			
METER			
SCROLLBAR			
LISTBOX			
LISTBOXEX			
CONTAINER			
FRAME			
LABEL	O	O	O
BAR	O	O	O

	TEXT_NORMAL	TEXT_BOLD	TEXT_DENT	TEXT_SHADOW
TEXT	O	O	O	O
RICHTEXT	O	O	O	O
LOG				
LOGTEXT	O	O	O	O
EDITBOX	O	O	O	O
TEXTBOX	O	O	O	O
BUTTON	O	O	O	O
RADIO	O	O	O	O
CHECKBOX	O	O	O	O
TEXTURE				
LINE				
RENDER				
ICON	O	O	O	O
RECASTICON	O	O	O	O
RENDERICON				
METER				
SCROLLBAR				
LISTBOX				
LISTBOXEX				
CONTAINER				
FRAME				
LABEL	O	O	O	O
BAR	O	O	O	O

	EDIT_BLIND	*EDIT_TYPE_**
TEXT		
RICHTEXT		
LOG		
LOGTEXT		
EDITBOX	O	O
TEXTBOX		
BUTTON		
RADIO		
CHECKBOX		
TEXTURE		
LINE		
RENDER		
ICON		
RECASTICON		
RENDERICON		
METER		
SCROLLBAR		
LISTBOX		
LISTBOXEX		
CONTAINER		
FRAME		
LABEL		
BAR		

	NOBOUNCES	SCROLL_UNLOCK	SCROLL_LOCK
TEXT			
RICHTEXT			
LOG	O	O	O
LOGTEXT			
EDITBOX	O	O	O
TEXTBOX	O	O	O
BUTTON			
RADIO			
CHECKBOX			
TEXTURE			
LINE			
RENDER			
ICON			
RECASTICON			
RENDERICON			
METER			
SCROLLBAR			
LISTBOX	O	O	O
LISTBOXEX	O	O	O
CONTAINER	O	O	O
FRAME			
LABEL			
BAR			

	ITEM_STACK_V/H	*SCROLLBAR_DISPLAY_**
TEXT		
RICHTEXT		
LOG		
LOGTEXT		
EDITBOX		
TEXTBOX		
BUTTON		
RADIO		
CHECKBOX		
TEXTURE		
LINE		
RENDER		
ICON		
RECASTICON		
RENDERICON		
METER		
SCROLLBAR	O	O
LISTBOX	O	
LISTBOXEX	O	
CONTAINER		
FRAME		
LABEL		
BAR		

12.6.1 About NOHIT/HIT

Each control has default of whether or not to hit.

	Default	NOHIT	HIT
TEXT	NOHIT		O
RICHTEXT	HIT	O	
LOG	HIT	O	
LOGTEXT	NOHIT	O	
EDITBOX	HIT	O	
TEXTBOX	HIT	O	
BUTTON	HIT	O	
RADIO	HIT	O	
CHECKBOX	HIT	O	
TEXTURE	NOHIT		O
LINE	NOHIT		O
RENDER	HIT	O	
ICON	HIT	O	
RECASTICON	HIT	O	
RENDERICON	HIT	O	
METER	NOHIT		O
SCROLLBAR	NOHIT		
LISTBOX	HIT	O	
LISTBOXEX	HIT	O	
CONTAINER	HIT	O	
FRAME	HIT	O	
LABEL	NOHIT		O
BAR	HIT	O	

CHAPTER

THIRTEEN

INDICES AND TABLES

- genindex
- modindex
- search

INDEX