# Nibiru XR System API Unity Developer Guide

**V1.0.1**

**Nanjing Ruiyue (Nibiru) Technology Co., Ltd.**

**Nanjing·May·2019**

# Legal Notices

Nanjing Ruiyue (Nibiru) Technology Co., Ltd.
Tel: +86 (025) 89635828
Address: Room 405, Building 4, Chuqiaocheng, No.57, Andemen Street, Yuhuatai District, Nanjing, Jiangsu Province, PRC
E-mail: support@inibiru.com
Web: http://dev.inibiru.com
Tech Support QQ Group: **464811686**

Any suggestions are welcomed.

# Document Updates

| **Version** | **Updates** |
|---|---|
| 1.0.0 | Initial version |
| 1.0.1 | Content Optimization |

# Content

In this document, you will learn how to use codes to set XR system interface SDK (Unity). Please read carefully.

Main steps include: initialize service and obtain system interface.

# 1 Initialization

```
    NibiruTaskApi.Init();
```

Initialization code is usually called in Start method of the script. See sample code:

```
    void Start () {
        NibiruTaskApi.Init();
    }
    Note: NibiruTaskInit.cs has been added to MainCamera.prefab.
```

# 2 Obtain System Interface

In XR interface SDK (Unity), interfaces to open system applications are provided. For example, opening or operating on video player, file manager, gallery, browser, settings, and peripheral drives.

**Open Video Player**

```
    Void OpenVideoPlayer(string path, int loop = Video.VIDEO_KEY_LOOP_ON,
int decode = Video.VIDEO_PARAMETERS_DECODE_HARDWARE, int mode =
Video.VIDEO_PARAMETERS_MODE_NORMAL, int type =
Video.VIDEO_PARAMETERS_TYPE_2D)；
```

path: the local path or network address of the video.

loop: whether to loop this video or not. Loop by default.

```
     //Define parameter VALUE: enable loop: no/yes
          public const int VIDEO_KEY_LOOP_OFF = 0;
         public const int VIDEO_KEY_LOOP_ON = 1;
```

decode: decoding method. Default mode: hardware decoding

```
     //Define parameter VALUE: decoding method, hardware/software
          public const int VIDEO_PARAMETERS_DECODE_HARDWARE = 0;
         public const int VIDEO_PARAMETERS_DECODE_SOFTWARE = 1;
```

mode: video play mode. Default mode: normal plane video

```
 //Define parameter VALUE: play mode: plane/panorama/ IMax Dome
 , corresponding Intent parameter
        public const int VIDEO_PARAMETERS_MODE_NORMAL = 0;
        public const int VIDEO_PARAMETERS_MODE_360 = 1;
        public const int VIDEO_PARAMETERS_MODE_180 = 2;
        public const int VIDEO_PARAMETERS_MODE_FULLDOME = 3;
```

type: video type. Default type: 2D video

```
 //Define parameter VALUE: play mode: 2D/3D. corresponding Intent
        public const int VIDEO_PARAMETERS_TYPE_2D = 0;
        public const int VIDEO_PARAMETERS_TYPE_3D = 1;
```

For detailed parameters and configuration info, please refer to the definition of Video in NibiruTaskConstants

Pause video

```
        public static void PauseVideoPlayer();
```

Continue playing video

```
        public static void ResumeVideoPlayer();
```

Close video player

```
        public static void CloseVideoPlayer();
```

Fast forward and backward

```
        public static void VideoPlayerSeekto(long time);
```

time: “long” type of time value for fast forward and backward

Display and hide video control bar

```
        public static void VideoPlayerShowORHideController(string state);
```

state: display or hide video control bar

```
 / /Define parameter VALUE: display control bar: yes/no
```

```
            public const string VIDEO_KEY_CONTROLLER_HIDE = "true";
            public const string VIDEO_KEY_CONTROLLER_SHOW = "false";
```

### Open File Manager

```
           public static void OpenFileExplorer(string path);
```

path: open file manager based on changing the path

```
           public static void OpenFileExplorer(int type);
```

type: open file manager based on file type

```
      //Define parameter VALUE: file type: video/image/apk
            public const int FILE_TYPE_VIDEO = 0;
            public const int FILE_TYPE_IMAGE = 1;
           public const int FILE_TYPE_APK = 2;
```

Obtain file path

```
     public static void GetFilePath(string basePath)
```

basePath: open file manager based on this path. Select the file to obtain its path.

Set the listener for the return value of the obtained file path:

```
      NibiruTaskApi.setSelectionCallback(onSelectionResult);
```

For processing listener, the returned code and path are required. Please refer to: NibiruTaskApi.GetResultValueFromSelectionTask(task);

```
      public void onSelectionResult(AndroidJavaObject task)
         {
            NibiruTaskApi.GetResultValueFromSelectionTask(task);
         }
```

### Open Gallery

```
          public static void OpenImageGallery(string path, int type =
Gallery.SHOW_IMAGE_KEY_2D);
```

path: the path of photo

type: image type. Default type: 2D Image

```
      //Define parameter VALUE: image format: 2D/3D/360 degree
            public const int SHOW_IMAGE_KEY_2D = 0;
            public const int SHOW_IMAGE_KEY_3D = 1;
```

```
        public const int SHOW_IMAGE_KEY_360 = 2;
```

### Open Settings

Open WIFI

```
        public static void OpenSettingsWifi();
```

Open Bluetooth

```
        public static void OpenSettingsBluetooth();
```

Open system settings

```
       public static void OpenSettingsSystem();
```

Open general settings

```
        public static void OpenSettingsGeneral();
```

Open main settings

```
        public static void OpenSettingsMain();
```

### Open Browser

```
        public static void OpenBrowerExplorer(string url, string actionBarState =
Brower.EXPLORER_KEY_ACTIONBAR_SHOW);
```

url: website address

actionBarState: display the address bar or not. Display by default.

```
    //Define parameter VALUE: display address bar: yes/no
        public const string EXPLORER_KEY_ACTIONBAR_HIDE = "true";
        public const string EXPLORER_KEY_ACTIONBAR_SHOW = "false";
```

# 3 Obtain System Info

In XR interface SDK (Unity), interfaces to open system applications are provided. For example, obtain channel code, device model number, system version number, service version code, brightness, and display mode.

**Obtain Channel Code**

```
public static string GetChannelCode();
```

**Obtain Device Model Number**

```
public static string GetModel();
```

**Obtain System Version Number**

```
public static string GetOSVersion();
public static int getOSVersionCode();
```

**Obtain Service Version Code**

```
public static string GetServiceVersionCode();
```

**Obtain Vendor Software Version**

```
public static string GetVendorSWVersion();
```

**Obtain Brightness**

```
public static int GetBrightnessValue();
```

Return value: 0~255 int type represents the current screen brightness

**Set brightness**

```
public static void SetBrightnessValue(int value);
```

Set value: 0~255 int type represents the set brightness value of the screen

**2D/3D Display Mode**

```
public static int GetDisplayMode();
```

Return value: 0=2D mode; 1=3D mode

### Set 2D/3D display mode

```
public static void SetDisplayMode(int mode);
```

Set mode value: 0=2D mode; 1=3D mode

### Obtain System Power Usage

System power is obtained by registered listener in XR interface. Users will be notified the power change by a callback.

Bind the service of setting before setting parameter:

```
NibiruTaskApi.addOnPowerChangeListener(onPowerChanged);
```

After the initialization, the callback of onPowerChanged(double value) will be initiated. Value represents the current system power ratio. The value of double type ranges from 0 to 1. The power can be processed through that callback function.

```
public void onPowerChanged(double value);
```

Cancel the registered listener when exit:

```
NibiruTaskApi.removeOnPowerChangeListener(onPowerChanged);
```

### Obtain Network Status

```
NibiruTaskApi.GetNetworkStatus();
```

0=no, 1=yes

### Obtain Bluetooth Status

```
NibiruTaskApi.GetBluetoothStatus();
```

0=turned off, 1=turned on

# 4 Obtain Setting Parameters

In XR interface SDK (Unity), interfaces to open system applications are provided. For example, obtain current theme, theme list, current time zone, language list, current language, and VR version info.

Bind the service of settings before obtaining setting parameter:

```
 NibiruTaskApi.addOnServerApiReadyCallback(onServerApiReady);
```

After initialization, onServerApiReady will be called back. Parameter isReady means it has been successfully bound. Parameters of settings can be obtained after binding.

```
public void onServerApiReady(bool isReady);
```

Cancel the registered listener and unbind the service for exit:

```
NibiruTaskApi.removeOnServerApiReadyCallback(onServerApiReady);
```

**Theme**

```
public static ThemeApiData GetCurrentTheme();
```

Return value: ThemeApiData type. Please refer to ThemeApiData for definitions

```
 private string themeName; //Theme name
 private string themeSign; //Theme sign
 private string themeIcon; //Theme icon
```

**Obtain all the theme lists**

```
public static List<ThemeApiData> GetThemeList();
```

Return value: List<ThemeApiData> type. Please refer to ThemeApiData for details

**Obtain Current Time Zone**

```
public static string GetCurrentTimezone();
```

**Obtain Language List**

```
public static List<string> GetLanguageList();
```

**Obtain Current Language**

```
public static string GetCurrentLanguage();
```

### Obtain Current AR Version Info

```
public static string GetVRVersion();
```

### Obtain Device Name

```
public static string GetDeviceName();
```

### Enable WPS

```
public static void OpenWps();
```

In XR interface SDK (Unity), interfaces to open system applications are provided= is provided by HMD service API SDK (Unity)

Bind the service of settings first before obtaining setting parameter:

```
NibiruTaskApi.addOnSysSleepApiReadyCallback(onSysSleepApiReady);
```

After initialization, the callback of onSysSleepApiReady is initiated. Parameter isReady means it has been successfully bound. The sleep time parameter can be obtained after binding.

```
public void onSysSleepApiReady(bool isReady);
```

Cancel the registered listener and unbind the service during exit:

```
NibiruTaskApi.removeOnSysSleepApiReadyCallback(onSysSleepApiReady);
```

### Obtain Sleep Time

```
public static int GetSysSleepTime();
```