

Instructions for using Token S DK

Table of contents

1. Training token data	2
1. Training must be performed before use, otherwise the SDK cannot be called	2
2. Use of training tools	2
2. Installation of dongle driver and operating environment	2
1. Install dongle driver	2
2. Install the operating environment	3
3. Dongle registration (required step before using SDK)	3
4. Desktop recognition SDK unity development and use documentation	5
1. Import and use of TouchScript plug-in	5
2. Import the dll of the development SDK into the Plugins folder	6
3. SDK case unity project code and CalculateTouch.cs script usage instructions	8
5. Instructions for using c/c++ to call SDK	12
1. How to call lib	12
2. Non-lib caller	13
6. Detailed explanation of error number	15
7. Unreal Engine SDK calling methods and cases	15

1. Training token data

1. Training must be performed before use, otherwise the S DK cannot be called.

2. Use of training tools

Just use the token training tool TokenData for training. Please see its instructions for specific instructions.

2. Installation of dongle driver and operating environment

1. Install the dongle driver

Open the dongle driver runner under the RunTime folder





Click Install Now and follow **the next step to complete the installation** according to the boot defaults .

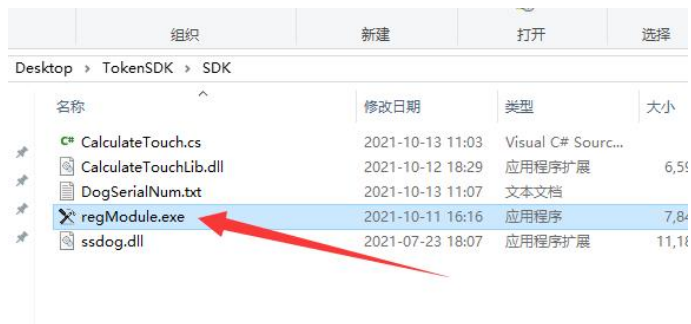
2. Install the operating environment

vcredist_x86 running program in the RunTime folder and complete the installation **in the next step according to the boot defaults** .

3. Dongle registration (required step before using SDK)

Dongle must be registered before development and testing

Registration steps: First run SDK\regModule.exe



As shown in the picture above , if you are connected to the Internet, please click online to register. If you are unable to connect to the Internet, please click

To register manually, send the serial number to the manufacturer to obtain the registration code . A computer only requires

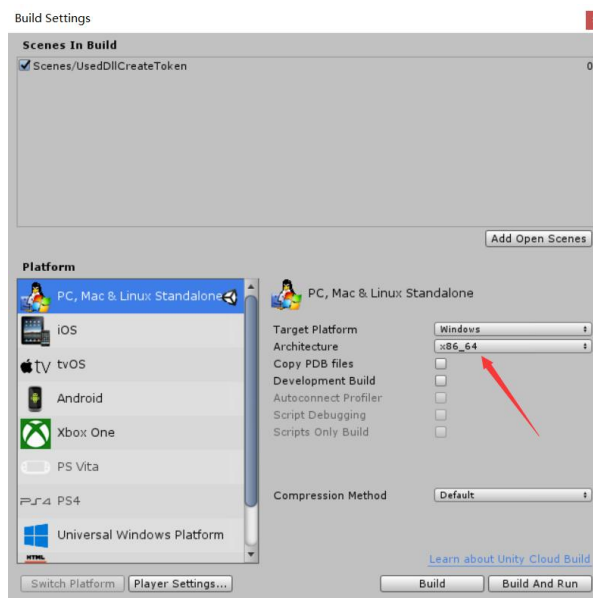
Register once. If you change the computer, you need to register again .

DK folder must be released to the customer when publishing , and the client computer needs to be registered once using the registration tool in the S DK .

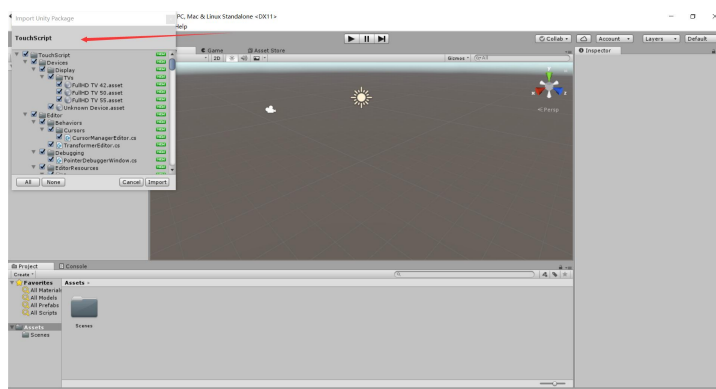
4. Desktop recognition SDK Unity development and use documentation

1. Import and use of TouchScript plug-in

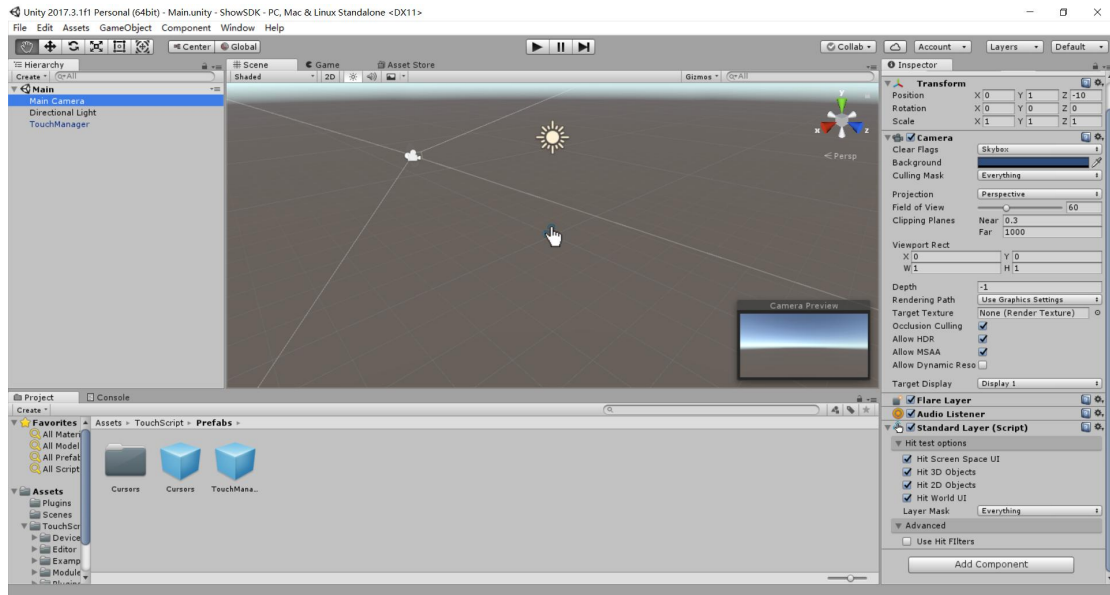
Step 1 : Set the project to run at 6 4



Part 2: Import the TouchScript plug-in



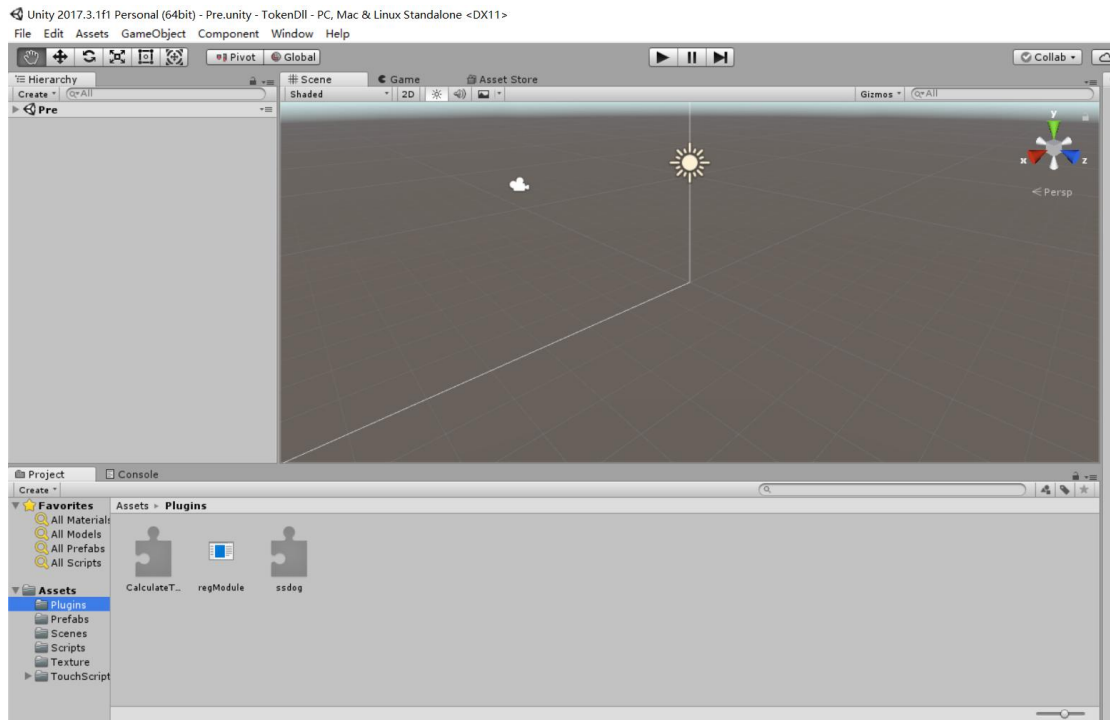
Step 3: Use the plug-in



Drag the TouchManager prefab in the TouchScript plug-in into the scene and add the StandardLayer component to the camera. Then you can get the token information on the screen by calling the CalculateTouch.cs script placed in Plugins in the previous step .

2. Import the dll of the development SDK into the Plugins folder

名称	修改日期
CalculateTouch.cs	2021-10-13 11:03
CalculateTouchLib.dll	2021-10-12 18:29
regModule.exe	2021-10-11 16:16
ssdog.dll	2021-07-23 18:07



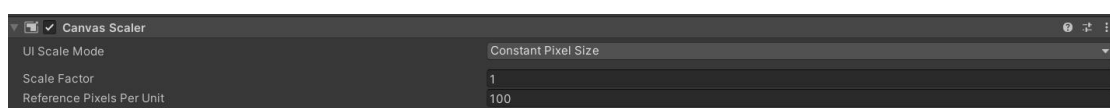
from the SDK folder into the project's Plugins folder

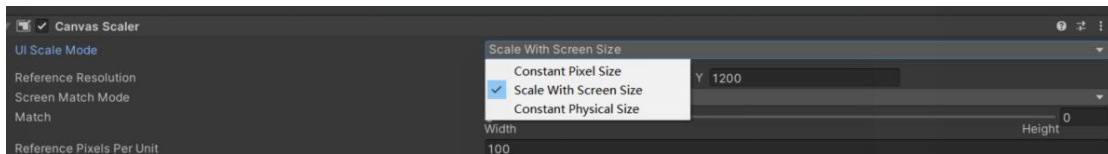
3、 UI Settings Precautions

The default development of UnityUI Canvas in Canvas Scaler mode

1 The Constant Pixel Size mode is compatible with token data at any resolution.

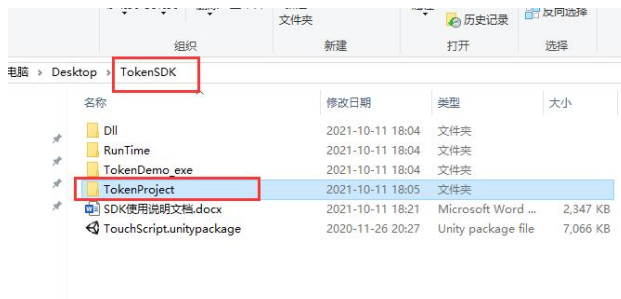
2 To use the Scale With Screen Size mode, you need to set the Reference Resolution to the corresponding resolution of the actual operation. For example, when running on a 3840 2160 screen, you can set the Reference Resolution to 3840 2160.





3. S DK case unity project code and CalculateTouch .cs script usage instructions

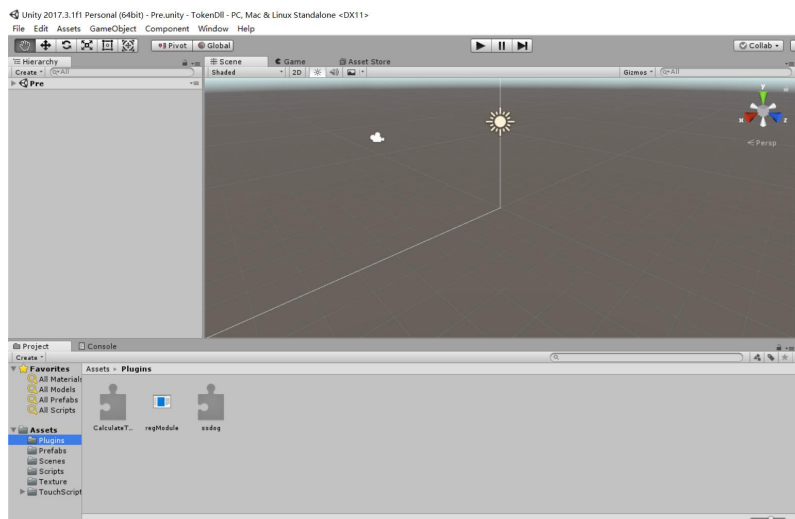
1. S DK case project is unity 2017.3.1 version. The project path is as shown in the figure below



2. Use of CalculateTouch.cs script interface in SDK

Step 1: Import SDK files and script CalculateTouch .cs

Copy all files in the S DK folder to unity In the Plugins folder under the project, as shown in the figure below



Step 2: Definition of touch plug-in interface

Define the interface in the Start interface

```
// Use this for initialization
@ Unity 消息 10 个引用
void Start () {
    if (TouchManager.Instance != null)
    {
        TouchManager.Instance.PointersAdded += pointersAddedHandler;
        TouchManager.Instance.PointersRemoved += pointersRemovedHandler;
        TouchManager.Instance.PointersPressed += pointersPressedHandler;
        TouchManager.Instance.PointersReleased += pointersReleasedHandler;
        TouchManager.Instance.PointersUpdated += PointersUpdatedHandler;
        TouchManager.Instance.PointersCancelled += pointersCancelledHandler;
    }
}
```

Step 3: Initialize the interface call

the initTokenData interface in the Start interface

Parameter 1: Container for returned data

: Path to ConfigSize.xml file

Parameter 3: Screen size

Parameter 4: Number of tokens to use (preferably read data set in external xml)

```
string xmlpath = Application.dataPath.Substring(0, Application.dataPath.LastIndexOf("/") + "/Resources/ConfigSize.xml");
byte[] info = new byte[1024];
if (CalculateTouch.InitTokenData(info, Encoding.UTF8.GetBytes(xmlpath), PreLoadSize.Instance.SelectSize, ReadConfig.Readtokencount()) != 0)
{
    timetip.SetActive(true);
    //错误码: 1001 未安装加密狗驱动. 1002 未检测到加密狗. 1003 加密狗到期 1004 加密狗到期, 请联系厂家
    if (Convert.ToInt32(Encoding.UTF8.GetString(info).Substring(6)) == 1001)
    {
        infoText.text = "未安装加密狗驱动";
    }
    else if (Convert.ToInt32(Encoding.UTF8.GetString(info).Substring(6)) == 1002)
    {
        infoText.text = "未检测到加密狗";
    }
    else if (Convert.ToInt32(Encoding.UTF8.GetString(info).Substring(6)) == 1003)
    {
        infoText.text = "加密狗到期, 请联系厂家";
    }
    else if (Convert.ToInt32(Encoding.UTF8.GetString(info).Substring(6)) == 1004)
    {
        infoText.text = "加密狗到期, 请联系厂家";
    }
}
```

Step 4: Calling the touch point data interface

Call the change of touch point information in the touch plug-in delegate interface (see step 1)

```
//在对应的触屏事件时传入点的信息
1 个引用
private void pointersAddedHandler(object sender, PointerEventArgs e)
{
    string info="";
    for (int i = 0; i < e.Pointers.Count; i++)
    {
        var pointer = e.Pointers[i];
        info += pointer.Id + "," + pointer.Position.x + "," + pointer.Position.y + "|";
    }

    CalculateTouch.pressDownTouchPt(Encoding.UTF8.GetBytes(info.ToCharArray()));
}

3 个引用
private void pointersRemovedHandler(object sender, PointerEventArgs e)
{
    string info = "";
    for (int i = 0; i < e.Pointers.Count; i++)
    {
        var pointer = e.Pointers[i];
        info += pointer.Id + "," + pointer.Position.x + "," + pointer.Position.y + "|";
    }

    CalculateTouch.removeUpTouchPt(Encoding.UTF8.GetBytes(info.ToCharArray()));
}
```

```
1 个引用
private void pointersPressedHandler(object sender, PointerEventArgs e)
{
    string info = "";
    for (int i = 0; i < e.Pointers.Count; i++)
    {
        var pointer = e.Pointers[i];
        info += pointer.Id + "," + pointer.Position.x + "," + pointer.Position.y + "|";
    }

    CalculateTouch.pressDownTouchPt(Encoding.UTF8.GetBytes(info.ToCharArray()));
}

1 个引用
private void PointersUpdatedHandler(object sender, PointerEventArgs e)
{
    string info = "";
    for (int i = 0; i < e.Pointers.Count; i++)
    {
        var pointer = e.Pointers[i];
        info += pointer.Id + "," + pointer.Position.x + "," + pointer.Position.y + "|";
    }

    CalculateTouch.updateMoveTouchPt(Encoding.UTF8.GetBytes(info.ToCharArray()));
}
```

```

1 个引用
private void pointersReleasedHandler(object sender, PointerEventArgs e)
{
    pointersRemovedHandler(sender, e);
}

1 个引用
private void pointersCancelledHandler(object sender, PointerEventArgs e)
{
    pointersRemovedHandler(sender, e);
}

```

Step 5: Calling and data analysis of token data interface

Implement the getTouchInfo interface in Update

```

// Update is called once per frame
void Update () {
    //通过dll的接口获取到令牌的信息
    byte[] info = new byte[1024];
    CalculateTouch.getTouchInfo(info);
    string str = Encoding.UTF8.GetString(info, 0, info.Length);
    UpdateToken(str);
}

```

The parsed code is in the UpdateToken interface in the TestDll.cs script

Obtained data Info The data format is as follows:

I D,X,Y,A ngle | I D,X,Y,A ngle | I D,X,Y,A ngle | I D,X,Y,A ngle

| : is the separator

ID : ID of the token

X, Y : The center coordinate position of the token

Angle : The incremental rotation angle of the current token (the rotation angle compared with the previous frame)

Step 6: Release of dll resources, called when the software exits

Call clearToken() in OnApplicationQuit() interface

```
86
87 #if !UNITY_EDITOR
88     void OnApplicationQuit()
89     {
90         CalculateTouch.clearToken();
91     }
92 #endif
93
```

Step Seven: Release Package

The C onfigSize .xml file is a necessary file for token recognition and must be packaged into the release package. You can refer to the case project and put it in the resources folder. All files under the S DK file need to be copied to the corresponding Plugins folder

DK folder must be released to the customer when publishing , and the client computer needs to be registered once using the registration tool in the S DK .

5. Instructions for calling S DK from c /c++

1. How to call lib

First, copy CalculateTouchLib.lib and CalculateTouchLib.dll to the current directory of the development project, then directly reference the header file of CalculateTouchLib.h in the code , and then call the corresponding interface

Step 1 : Call the initialization token parameters and desktop information. You only need to

execute it once.

```
/*
初始化 只需要调用一次
返回0: 成功 非0: 错误 错误码存放在lpbuffIn字节中
错误码:
1001 未安装加密狗驱动.
1002 未检测到加密狗.
1003 加密狗到期
1004 加密狗到期, 请联系厂家
1005 令牌参数不对
1006 xml配置文件出错或丢失
1007 无令牌模型数据
参数一: 返回的错误代码数组容器
参数二: 配置文件的configsize.xml 配置数据由我方提供, 不同屏幕尺寸数据不一样
参数三: 屏幕尺寸 43寸 55寸 65寸 75寸
参数四: 令牌数量
*/
CALCULATETOUCHLIB_API int initTokenData(BYTE *lpbuffIn, BYTE *lpXml, int nSize, int nTokenCount);
```

Step 2 : Add the interface call corresponding to the touch message

```
//触摸屏事件 按下 返回0: 成功 非0: 错误
//参数: 触摸点的ID号, X,Y坐标 数据格式: id,x,y|id,x,y|....
CALCULATETOUCHLIB_API int pressDownTouchPt(BYTE *lpbuffIn);

//触摸屏事件 移动(更新) 返回0: 成功 非0: 错误
//参数: 触摸点的ID号, X,Y坐标 数据格式: id,x,y|id,x,y|....
CALCULATETOUCHLIB_API int updateMoveTouchPt(BYTE *lpbuffIn);

//触摸屏事件 抬起(删除) 返回0: 成功 非0: 错误
//参数: 触摸点的ID号, X,Y坐标 数据格式: id,x,y|id,x,y|....
CALCULATETOUCHLIB_API int removeUpTouchPt(BYTE *lpbuffIn);
```

Step 3 : Use getTouchInfo to obtain the token information, which can be called according to the number of refresh rates of the software

```
//解析点数据 每帧调用一次 返回0: 成功 非0: 错误
//参数: 获取到圆形物理令牌的信息数组容器 容器大小2048个字节
//数据格式(令牌ID 令牌xy坐标, 令牌的增量旋转角度): ID,X,Y,Angle|ID,X,Y,Angle|ID,X,Y,Angle|....
CALCULATETOUCHLIB_API int getTouchInfo(BYTE* lpbuffOut);
```

Step 4 : Release of dll resources, called when the software exits

```
//退出软件前调用, 释放所有资源(必须调用)
CALCULATETOUCHLIB_API void clearToken();
```

2. Non-lib callers

First copy CalculateTouchLib.dll to the current directory of the development project, and then directly reference the header file of CalculateTouchC.h in the code

The first step : initialize the dll function and obtain all function pointer addresses

```
//获取库函数
void initDllFunc()
```

Step 2 : Call the initialization token parameters and desktop information. You only need to execute it once.

```
/*
初始化 只需要调用一次
返回0: 成功 非0: 错误 错误码存放在lpbuffIn字节中
错误码:
1001 未安装加密狗驱动.
1002 未检测到加密狗.
1003 加密狗到期
1004 加密狗到期, 请联系厂家
1005 令牌参数不对
1006 xml配置文件出错或丢失
1007 无令牌模型数据
参数一: 返回的错误代码数组容器
参数二: 配置文件的configsize.xml 配置数据由我方提供, 不同屏幕尺寸数据不一样
参数三: 屏幕尺寸 43寸 55寸 65寸 75寸
参数四: 令牌数量
*/
FuncInitTokenData g_funcInitTokenData;
```

Step 3 : Add the interface call corresponding to the touch message

```
//触摸屏事件 按下 返回0: 成功 非0: 错误
//参数: 触摸点的ID号, X,Y坐标 数据格式: id,x,y|id,x,y|.....
FuncPressDownTouchPt g_funcPressDownTouchPt;

//触摸屏事件 移动(更新) 返回0: 成功 非0: 错误
//参数: 触摸点的ID号, X,Y坐标 数据格式: id,x,y|id,x,y|.....
FuncRemoveUpTouchPt g_funcRemoveUpTouchPt;

//触摸屏事件 抬起(删除) 返回0: 成功 非0: 错误
//参数: 触摸点的ID号, X,Y坐标 数据格式: id,x,y|id,x,y|.....
FuncUpdateMoveTouchPt g_funcUpdateMoveTouchPt;
```

Step 4 : Use getTouchInfo to obtain the token information, which can be called according to the number of refresh rates of the software

```
//解析点数据 每帧调用一次 返回0: 成功 非0: 错误
//参数: 获取到圆形物理令牌的信息数组容器
//数据格式(令牌ID 令牌xy坐标, 令牌的增量旋转角度): ID,X,Y,Angle|ID,X,Y,Angle|ID,X,Y,Angle|.....
FuncGetTouchInfo g_funcGetTouchInfo;
```

Step 5 : Release of token resources, release of dll, called when the software exits

```
//关闭释放token资源
FuncClearToken g_funcClearToken;

//关闭库
void delInit()
```

6. Detailed explanation of error number

1001 The dongle driver is not installed .

1002 Dongle not detected .

1003 Dongle expires

1004 Dongle expires, please contact the manufacturer

1005 Token parameters are incorrect

1006 xml configuration file error or missing

1007 There is no token model data for this size screen. Please train the token first.

1008 configsize.xml configuration file does not exist

1009 configsize.xml configuration file format error

1010Screen size cannot be 0

1011 Token model data error

1012 Dongle not registered

1013 initTokenData Interface not called completed

7. Calling methods and cases of Unreal Engine using SDK

Please contact technical support to request