BS2_EnableDeviceLicense

SDK API > BS2_EnableDeviceLicense

BS2_EnableDeviceLicense

[+ 2.8.4]　　　　　　　　　　　　　　　　　　,
　．

　outResultObj　outNumOfResult　　　　　　．
　　　　　　　　　　　　　　　　　　　　　　，
　　．

XS2-Finger	V1.2.0
XS2-Card	V1.2.0
BS3	V1.1.0

```
#include "BS_API.h"

int BS2_EnableDeviceLicense(void* context, uint32_t deviceId, const
BS2LicenseBlob* licenseBlob, BS2LicenseResult** outResultObj, uint32_t*
outNumOfResult);
```

> BS2LicenseBlob
> BS2LicenseResult

- [In] *context* : Context
- [In] *deviceId* :
- [In] *licenseBlob* :
- [Out] *outResultObj* :
- [Out] *outNumOfResult* :

outResultObj BS2_ReleaseObject ．

BS_SDK_SUCCESS ,

.

(C++)

sample_setdebugfilelogex.cpp

```cpp
const char* CURRENT_DIR = ".";
const int MAX_SIZE_LOG_FILE = 100;  // 100MB
int sdkResult = BS2_SetDebugFileLogEx(DEBUG_LOG_ALL, DEBUG_MODULE_ALL,
CURRENT_DIR, MAX_SIZE_LOG_FILE);
if (BS_SDK_SUCCESS != sdkResult)
{
    printf("BS2_SetDebugFileLogEx call failed: %d", sdkResult);
    return;
}
```

(C#)

sample_setdebugfilelogex.cs

```csharp
const string CURRENT_DIR = ".";
const int MAX_SIZE_LOG_FILE = 100;  // 100MB
IntPtr ptrDir = Marshal.StringToHGlobalAnsi(CURRENT_DIR);
result =
(BS2ErrorCode)API.BS2_SetDebugFileLogEx(Constants.DEBUG_LOG_OPERATION_A
LL, Constants.DEBUG_MODULE_ALL, ptrDir, MAX_SIZE_LOG_FILE);
Marshal.FreeHGlobal(ptrDir);
if (result != BS2ErrorCode.BS_SDK_SUCCESS)
{
    Console.WriteLine("Got error({0}).", result);
    return;
}
```

From:
https://kb.supremainc.com/kbtest/ - **BioStar Device SDK**

Permanent link:
https://kb.supremainc.com/kbtest/doku.php?id=ko:bs2_enabledevicelicense&rev=1675927021

Last update: **2023/02/09 16:17**