

BS2_QueryDeviceLicense

.....

.....

.....

.....

(C++)

(C#)

1

1

1

2

2

2

BS2_QueryDeviceLicense

[+ 2.9.1]

/

가

.

가

,

.

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

```
#include "BS_API.h"

int BS2_QueryDeviceLicense(void* context, uint32_t deviceId,
BS2_LICENSE_TYPE licenseType, BS2LicenseResult** outResultObj, uint32_t*
outNumOfResult);
```

BS2LicenseResult

- [In] *context* : Context
- [In] *deviceId* :
- [In] *licenseType* :

0x0000	None
0x0001	Visual QR

- [Out] *outResultObj* :
- [Out] *outNumOfResult* :



BS_SDK_SUCCESS , 가

(C++)

[sample_setdebugfilelogex.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](#)

```
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:

<http://kb.supremainc.com/bs2sdk/> - **BioStar Device SDK**

Permanent link:

http://kb.supremainc.com/bs2sdk./doku.php?id=ko:bs2_querydevicelicense&rev=1676353731

Last update: **2023/02/14 14:48**