Table of Contents

BS2_EnableDeviceLicense

[+ 2.9.1] Activate the device license by selecting the devices to activate the license and the license file.
Slave devices connected to the master device can be collectively selected, and activation results for each device are returned through outResultObj and outNumOfResult.
This feature is available only on devices that support the device license activation feature, and the devices that support the feature are listed below.

Supported devices	Firmware
XS2-Finger	V1.2.0
XS2-Card	V1.2.0
BS3	V1.1.0

Declaration

```
#include "BS_API.h"

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

See BS2LicenseBlob Structure
See BS2LicenseResult Structure

Parameter

- [In] *context* : Context
- [In] *deviceId* : Device Identifier
- [In] *licenseBlob* : Device license information structure pointer
- [Out] *outResultObj* : Pointer to structure to receive device license activation result
- [Out] *outNumOfResult* : Number of device license activation result structures

> **NOTE**
> The outResultObj parameter must be used and then returned
> memory to the system via the BS2_ReleaseObject function.

Return Value

If successfully done, BS_SDK_SUCCESS will be returned. If there is an error, the corresponding error code will be returned.

Sample Code(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;
}
```

Sample Code (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;
}
```