

Table of Contents

- BS2_GetSocketRetryCount** 1
- Declaration 1
- Parameter 1
- Return Value 1
- See Also 1
- Sample Code(C++) 1
- Sample Code (C#) 2

[Communication API](#) > [BS2_GetSocketRetryCount](#)

BS2_GetSocketRetryCount

[+ 2.9.1] During normal TCP/IP socket communication, if an error occurs while reading and writing packets from the socket in the communication section, the SDK automatically retries a certain number of times.

This API gets the set value for the number of retries set in the SDK.

Declaration

```
#include "BS_API.h"

int BS2_GetSocketRetryCount(void* context, uint32_t* count);
```

Parameter

- [In] *context* : Context
- [Out] *count* : Retry count setting value pointer

Return Value

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

See Also

[BS2_SetSocketRetryCount](#)

Sample Code(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;
}
```

```
}
```

Sample Code (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:

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

Permanent link:

https://kb.supremainc.com/kbtest/doku.php?id=en:bs2_getsocketretrycount&rev=1677563499

Last update: **2023/02/28 14:51**