


Table of Contents

| | |
|------------------------------------|---|
| BS2_SetDebugFileLogEx | 1 |
| 함수 | 1 |
| 파라미터 | 1 |
| 반환값 | 2 |
| 샘플코드(C++) | 2 |
| 샘플코드(C#) | 3 |

 **Fix Me!** This page is not fully translated, yet. Please help completing the translation.
(remove this paragraph once the translation is finished)

[SDK API](#) > [BS2_SetDebugFileLogEx](#)

BS2_SetDebugFileLogEx

[+ 2.8.3] Log messages within the SDK can be output as a split file and used for debugging applications.

The fourth factor, fileMaxSizeMB, allows you to specify a maximum size per file, in MB.

If you set this value to 0, you will not proceed with file segmentation saving.

The file can be specified as either an absolute path or a relative path, and log messages that occur during subsequent SDK operations are automatically generated in that path.

The file name stored at this time is YYYYMMDD_x.Take the form of log (_x is given sequentially from 0 whenever fileMaxSize is exceeded)

함수

```
#include "BS_API.h"

int BS2_SetDebugFileLogEx(uint32_t level, uint32_t module, const char*
logPath, int fileMaxSizeMB);
```

파라미터

- [In] level : 오류 수준을 지정

| 매크로 | 값 | 의미 |
|-------------------------|------------|---------------------|
| DEBUG_LOG_FATAL | 0x00000001 | 치명적인 오류 |
| DEBUG_LOG_ERROR | 0x00000002 | 일반 오류 |
| DEBUG_LOG_WARN | 0x00000004 | 경고 |
| DEBUG_LOG_API | 0x00000008 | API 호출 IN 및 OUT |
| DEBUG_LOG_INFO | 0x00000010 | 오류 외의 정보 |
| DEBUG_LOG_TRACE | 0x00000100 | SDK 자체 디버깅 목적의 정보 |
| DEBUG_LOG_SYSTEM | 0x0000000F | 모든 오류 수준을 출력 |
| DEBUG_LOG_OPERATION_ALL | 0x000000FF | 모든 오류 수준과 일반 정보를 출력 |

| 매크로 | 값 | 의미 |
|---------------|------------|----------|
| DEBUG_LOG_ALL | 0xFFFFFFFF | 모든 정보 출력 |

- [In] *module*: 모듈을 지정

| 매크로 | 값 | |
|-------------------------------|------------|---------------------|
| DEBUG_MODULE_KEEP_ALIVE | 0x00000001 | Keep alive 모듈 |
| DEBUG_MODULE_SOCKET_MANAGER | 0x00000002 | 소켓 관리 모듈 |
| DEBUG_MODULE_SOCKET_HANDLER | 0x00000004 | 소켓 운영 모듈 |
| DEBUG_MODULE_DEVICE | 0x00000008 | 장치 모듈 |
| DEBUG_MODULE_DEVICE_MANAGER | 0x00000010 | 장치 관리 모듈 |
| DEBUG_MODULE_EVENT_DISPATCHER | 0x00000020 | 이벤트 처리기 모듈 |
| DEBUG_MODULE_API | 0x00000040 | API |
| DEBUG_MODULE_MISC | 0x00000080 | 기타 |
| DEBUG_MODULE_PACKET | 0x00000100 | 통신 패킷 처리 모듈 |
| DEBUG_MODULE_NOTIFY_MANAGER | 0x00000400 | Notify 처리 모듈 |
| DEBUG_MODULE_EVENT | 0x00000800 | USB event log 처리 모듈 |
| DEBUG_MODULE_USB | 0x00001000 | USB import 처리 모듈 |
| DEBUG_MODULE_ALL | 0xFFFFFFFF | 전체 모듈 |

- [In] *logPath*: 로그파일이 출력될 경로
- [In] *fileMaxSizeMB*: 저장할 파일의 최대크기

반환값

성공적으로 수행될 경우 BS_SDK_SUCCESS를 반환하고, 에러가 발생할 경우 상응하는 에러 코드를 반환합니다.

샘플코드(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;
}
```

샘플코드(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 2 Device SDK**

Permanent link:

https://kb.supremainc.com/kbtest/doku.php?id=en:bs2_setdebugfilelogex&rev=1662524771Last update: **2022/09/07 13:26**