

**BS2\_GetLogSmallBlobEx** ..... 1  
..... 1  
..... 1  
..... 1  
..... 1

# BS2\_GetLogSmallBlobEx

[+ V2.7.1] EventMask

가

```
#include "BS_API.h"

int BS2_GetLogSmallBlobEx(void* context, uint32_t deviceId, uint16_t
eventMask, uint32_t eventId, uint32_t amount, BS2EventSmallBlobEx** logsObj,
uint32_t* numLog);
```

[BS2EventSmallBlobEx](#)

- [In] *context* : Context
- [In] *deviceId* :
- [In] *eventMask* : event mask
- [In] *eventId* : ID , 0 가 .
- [In] *amount* : , 0 eventId 가 .
- [Out] *logsObj* :
- [Out] *numLog* :

logsObj [BS2\\_ReleaseObject](#)

BS\_SDK\_SUCCESS , 가

C++

```
int LogControl::getLogSmallBlobEx(BS2_DEVICE_ID id)
{
    uint16_t mask = BS2_EVENT_MASK_USER_ID | BS2_EVENT_MASK_TEMPERATURE;
    const uint32_t FROM_FIRST = ;
    BS2_EVENT_ID eID = FROM_FIRST;
    uint32_t amount(), numLog();
    BS2EventSmallBlobEx* blobObj = NULL;

    int sdkResult = BS2_GetLogSmallBlobEx(context_, id, mask, eID, amount,
    &blobObj, &numLog);
    if (BS_SDK_SUCCESS != sdkResult)
        TRACE("BS2_GetLogSmallBlobEx call failed: %d", sdkResult);

    print(blobObj, numLog);

    if (blobObj)
    {
        if (blobObj->imageObj)
            BS2_ReleaseObject(blobObj->imageObj);

        BS2_ReleaseObject(blobObj);
    }
    return sdkResult;
}
```

C#

```
outEventLogObjs = IntPtr.Zero;
BS2ErrorCode result = (BS2ErrorCode)API.BS2_GetLogSmallBlobEx(sdkContext,
deviceID, (ushort)BS2EventMaskEnum.ALL, lastEventId, amount, out
outEventLogObjs, out outNumEventLogs);
if (result != BS2ErrorCode.BS_SDK_SUCCESS)
{
    Console.WriteLine("Got error({0}).", result);
    break;
}
```

From:

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

Permanent link:

[https://kb.supremainc.com/bs2sdk/doku.php?id=ko:bs2\\_getlogsmallblobex](https://kb.supremainc.com/bs2sdk/doku.php?id=ko:bs2_getlogsmallblobex)

Last update: **2021/12/30 13:07**