

BS2_StartMonitoringLogEx

.....

.....

.....

.....

.....

.....

.....

1

1

1

1

1

1

2

[Log Management API](#) > [BS2_StartMonitoringLogEx](#)

BS2_StartMonitoringLogEx

[+ V2.7.1]

```
#include "BS_API.h"
```

```
int BS2_StartMonitoringLogEx(void* context, uint32_t deviceId,
OnLogReceivedEx ptrLogReceivedEx);
```

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

BS_SDK_SUCCESS , 가

C++

```
static DeviceInfo deviceInfo = { , , , 51211, };
static void* sdkContext = NULL;
sdkContext = BS2Context::getInstance()->getContext();

void onLogReceivedEx(BS2_DEVICE_ID id, const BS2Event* event,
BS2_TEMPERATURE temperature)
{
    if (deviceInfo.id_ == id)
    {
        int32_t timezone = deviceInfo.timezone_;
        cout << Utility::getEventStringWithThermal(id, *event, timezone,
temperature) << endl;
    }
}

sdkResult = BS2_StartMonitoringLogEx(sdkContext, deviceInfo.id_,
```

```
onLogReceivedEx);  
if (BS_SDK_SUCCESS != sdkResult)  
    TRACE("BS2_StartMonitoringLog call failed: %d", sdkResult);
```

C#

```
private API.OnLogReceived cbOnLogReceived = null; //To prevent garbage  
collection  
  
BS2ErrorCode result = (BS2ErrorCode)API.BS2_StartMonitoringLogEx(sdkContext,  
deviceID, cbOnLogReceivedEx);  
if (result != BS2ErrorCode.BS_SDK_SUCCESS)  
{  
    Console.WriteLine("Got error({0}).", result);  
}
```

BS2_StopMonitoringLog

From:

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

Permanent link:

https://kb.supremainc.com/bs2sdk/doku.php?id=ko:bs2_startmonitoringlogex

Last update: **2021/12/30 12:47**