

**BS2\_GetUserDatasFaceEx** ..... 1

..... 1

..... 1

..... 2

..... 2

..... 3

# BS2\_GetUserDatasFaceEx

[+ 2.7.1]

userMask  
Visual camera

가

가

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

```
int BS2_GetUserDatasFaceEx(void* context, uint32_t deviceId, char* uids,
uint32_t uidCount, BS2UserFaceExBlob* userBlob, BS2_USER_MASK userMask);
```

[BS2UserFaceExBlob](#)

- [In] *context* : Context
- [In] *deviceId* :
- [In] *uids* : 가
- [In] *uidCount* :
- [Out] *userBlob* :
- [In] *userMask* : Mask

|        |                  |
|--------|------------------|
| 0x0000 |                  |
| 0x0001 | data             |
| 0x0002 |                  |
| 0x0004 |                  |
| 0x0008 |                  |
| 0x0010 | PIN              |
| 0x0020 |                  |
| 0x0040 |                  |
| 0x0080 |                  |
| 0x0100 |                  |
| 0x0200 |                  |
| 0x0400 |                  |
| 0x0800 | (FaceStation F2) |
| 0x1000 | (FaceStation F2) |
| 0xFFFF |                  |

BS\_SDK\_SUCCESS , 가

C++

```

const int MAX_USER_PAGE_COUNT = 2048;
vector<BS2UserFaceExBlob> userBlob(MAX_USER_PAGE_COUNT);
BS2_USER_MASK userMask = BS2_USER_MASK_ALL;

for (uint32_t idx = ; idx < numUID; idx += MAX_USER_PAGE_COUNT)
{
    uint32_t availUserCount = min<uint32_t>(MAX_USER_PAGE_COUNT, numUID -
idx);

    int sdkResult = BS2_GetUserDatasFaceEx(context_, id, uidObj +
BS2_USER_ID_SIZE * idx, availUserCount, &userBlob[], userMask);
    if (BS_SDK_SUCCESS != sdkResult)
    {
        TRACE("BS2_GetUserDatas call failed: %d", sdkResult);
        if (uidObj)
            BS2_ReleaseObject(uidObj);
        return sdkResult;
    }

    for (uint32_t pageIdx = ; pageIdx < availUserCount; pageIdx++)
    {
        print(userBlob[pageIdx]);

        if ( < userBlob[pageIdx].user.numCards &&
userBlob[pageIdx].cardObjjs)
            BS2_ReleaseObject(userBlob[pageIdx].cardObjjs);
        if ( < userBlob[pageIdx].user.numFingers &&
userBlob[pageIdx].fingerObjjs)
            BS2_ReleaseObject(userBlob[pageIdx].fingerObjjs);
        if ( < userBlob[pageIdx].user.numFaces &&
userBlob[pageIdx].faceExObjjs)
            BS2_ReleaseObject(userBlob[pageIdx].faceExObjjs);
    }
}

```

C#

```

BS2UserFaceExBlob[] userBlobs = new BS2UserFaceExBlob[numUser];
BS2ErrorCode result = (BS2ErrorCode)API.BS2_GetUserDatasFaceEx(sdkContext,
deviceID, uid, numUser, userBlobs, userMask);

```

```
if (result == BS2ErrorCode.BS_SDK_SUCCESS)
{
    for (UInt32 index = ; index < numUser; index++)
    {
        print(userBlobs[index]);

        if (userBlobs[index].cardObjs != IntPtr.Zero)
            API.BS2_ReleaseObject(userBlobs[index].cardObjs);
        if (userBlobs[index].fingerObjs != IntPtr.Zero)
            API.BS2_ReleaseObject(userBlobs[index].fingerObjs);
        if (userBlobs[index].faceObjs != IntPtr.Zero)
            API.BS2_ReleaseObject(userBlobs[index].faceObjs);
        if (userBlobs[index].faceExObjs != IntPtr.Zero)
            API.BS2_ReleaseObject(userBlobs[index].faceExObjs);
    }
}
```

[BS2\\_EnrollUserFaceEx](#)  
[BS2\\_GetUserInfosFaceEx](#)  
[BS2\\_GetUserDatasFaceEx](#)

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

Permanent link:  
[https://kb.supremainc.com/bs2sdk/doku.php?id=ko:bs2\\_getuserdatasfaceex&rev=1716882960](https://kb.supremainc.com/bs2sdk/doku.php?id=ko:bs2_getuserdatasfaceex&rev=1716882960)

Last update: **2024/05/28 16:56**