

BS2_ScanFace 1
..... 1
..... 1
..... 1
..... 1

BS2_ScanFace

FaceStation2

```
#include "BS_API.h"

int BS2_ScanFace(void* context, uint32_t deviceId, BS2Face* face, uint8_t
erollmentThreshold, OnReadyToScan ptrReadyToScan);
```

BS2Face

- [In] *context* : Context
- [In] *deviceId* :
- [Out] *face* :
- [In] *erollmentThreshold* : erollmentThreshold - [BS2FaceConfig.enrollThreshold](#)
- [Out] *ptrReadyToScan* : 가

BS_SDK_SUCCESS , 가

C++

```
if (faceScanSupported)
{
    if (Utility::isYes("Do you want to scan face?"))
    {
        uint32_t numFace = Utility::getInput<uint32_t>("How many face would
you like to register?");
        BS2Face* ptrFace = new BS2Face[numFace];
        if (ptrFace)
```

```

    {
        userBlob.faceObjs = ptrFace;
        for (uint32_t index = ; index < numFace;)
        {
            sdkResult = BS2_ScanFace(context_, id, &ptrFace[index],
BS2_FACE_ENROLL_THRESHOLD_DEFAULT, onReadyToScanFace);
            if (BS_SDK_SUCCESS != sdkResult)
                TRACE("BS2_ScanFace call failed: %d", sdkResult);
            else
            {
                user.numFaces++;
                index++;
            }
        }
    }
}
BS2_ReleaseObject(uidObj);

```

C#

```

private API.OnReadyToScan cbFaceOnReadyToScan = null;
if (faceScanSupported)
{
    int numOfFace = 1;
    if (< numOfFace)
    {
        int structSize = Marshal.SizeOf(typeof(BS2FaceExWarped));
        BS2FaceExWarped[] faceEx =
Util.AllocateStructureArray<BS2FaceExWarped>(1);
        userBlob[].faceExObjs = Marshal.AllocHGlobal(structSize *
numOfFace);
        IntPtr curFaceExObjs = userBlob[].faceExObjs;
        cbFaceOnReadyToScan = new API.OnReadyToScan(ReadyToScanForFace);

        for (int index = ; index < numOfFace;)
        {
            sdkResult = (BS2ErrorCode)API.BS2_ScanFaceEx(context, deviceId,
faceEx, (byte)BS2FaceEnrollThreshold.THRESHOLD_DEFAULT,
cbFaceOnReadyToScan);
            if (BS2ErrorCode.BS_SDK_SUCCESS != sdkResult)
                Console.WriteLine("BS2_ScanFaceEx call failed: %d",
sdkResult);
            else
            {
                userBlob[].user.numFaces++;
                index++;
                faceEx[].faceIndex = (byte)index;
                Marshal.StructureToPtr(faceEx[], curFaceExObjs, false);
                curFaceExObjs += structSize;
            }
        }
    }
}

```

```
        Thread.Sleep(100);
    }
}
cbFaceOnReadyToScan = null;
}
BS2_ReleaseObject(uidObj);
```

From:

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

Permanent link:

http://kb.supremainc.com/bs2sdk/doku.php?id=ko:bs2_scanface&rev=1653465932

Last update: **2022/05/25 17:05**