

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#

```

if (faceScanSupported)
{
    Console.WriteLine("Do you want to scan face? [y/n]");
    Console.Write(">> ");
    if (Util.IsYes())
    {
        Console.WriteLine("How many face would you like to register?");
        Console.Write(">> ");
        int numOfFace = Util.GetInput(1);
        if (< numOfFace)
        {
            int structSize = Marshal.SizeOf(typeof(BS2Face));
            BS2Face[] face = Util.AllocateStructureArray<BS2Face>(1);
            userBlob[].faceObjs = Marshal.AllocHGlobal(structSize *
numOfFace);
            IntPtr curFaceObjs = userBlob[].faceObjs;
            cbFaceOnReadyToScan = new API.OnReadyToScan(ReadyToScanForFace);

            for (int index = ; index < numOfFace;)
            {
                sdkResult = (BS2ErrorCode)API.BS2_ScanFace(sdkContext,
deviceID, face, (byte)BS2FaceEnrollThreshold.THRESHOLD_DEFAULT,
cbFaceOnReadyToScan);
                if (BS2ErrorCode.BS_SDK_SUCCESS != sdkResult)
                    Console.WriteLine("BS2_ScanFace call failed: %d",
sdkResult);
                else
                {
                    userBlob[].user.numFaces++;
                }
            }
        }
    }
}

```

```
        index++;
        face[].faceIndex = (byte)index;
        Marshal.StructureToPtr(face[], curFaceObj, false);
        curFaceObj += structSize;

        Thread.Sleep(100);
    }
}

cbFaceOnReadyToScan = null;
}
}
```

From:

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

Permanent link:

http://kb.supremainc.com/bs2sdk/doku.php?id=ko:bs2_scanface&rev=1656460215

Last update: **2022/06/29 08:50**