목차

BS2_ScanFace 1
함수 1
파라미터 1
반환값 1
샘플코드 1

BS2_ScanFace

FaceStation2 장치에서 얼굴을 스캔하고 템플릿 데이터와 이미지 데이터를 추출합니다.

함수

```
#include "BS_API.h"

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

> BS2Face구조체 보기

파라미터

- [In] *context* : Context
- [In] *deviceId* : 장치 식별자
- [Out] *face* : 얼굴 데이터를 저장할 포인터
- [In] *erollmentThreshold* : 요구되는 얼굴 erollmentThreshold - 상세한 것은 BS2FaceConfig.enrollThreshold 참조
- [Out] *ptrReadyToScan* : 얼굴 스캔 준비가 완료되었을 때 호출되는 콜백 함수

반환값

성공적으로 수행될 경우 BS_SDK_SUCCESS를 반환하고, 에러가 발생할 경우 상응하는 에러 코드를 반환합니다.

샘플코드

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);
```