목차

BS2_GetUserDatasFaceEx

[+ 2.7.1] 사용자 식별자에 해당하는 사용자 정보를 가져옵니다.
이때 userMask를 통하여 사용자 정보의 일부를 선별적으로 가져올 수 있습니다.
이 함수는 Visual Face 인증 수단을 지원합니다.

함수

```
#include "BS_API.h"

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

> BS2UserFaceExBlob 구조체 보기

파라미터

- [In] *context* : Context
- [In] *deviceId* : 장치 식별자
- [In] *uids* : 가져오고자 하는 사용자 식별자 리스트
- [In] *uidCount* : 사용자 식별자 개수
- [Out] *userBlob* : 사용자 정보를 저장할 포인터
- [In] *userMask* : 사용자 Mask

값	설명
0x0000	사용자 식별자
0x0001	사용자 data
0x0002	사용자 설정
0x0004	사용자명
0x0008	이미지
0x0010	PIN
0x0020	카드
0x0040	지문
0x0080	얼굴
0x0100	출입그룹
0x0200	작업코드
0x0400	개인메시지
0x0800	얼굴 (Visual Face 지원)
0x1000	사용자설정 (Visual Face 지원)
0xFFFF	사용자 정보 전체

반환값

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

샘플코드

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].cardObjs)
            BS2_ReleaseObject(userBlob[pageIdx].cardObjs);
        if ( < userBlob[pageIdx].user.numFingers &&
userBlob[pageIdx].fingerObjs)
            BS2_ReleaseObject(userBlob[pageIdx].fingerObjs);
        if ( < userBlob[pageIdx].user.numFaces &&
userBlob[pageIdx].faceExObjs)
            BS2_ReleaseObject(userBlob[pageIdx].faceExObjs);
    }
}
```

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);
    }
}
```

함께 보기

BS2_EnrollUserFaceEx
BS2_GetUserInfosFaceEx
BS2_GetUserDatasFaceEx