

**BS2\_GetNormalizedImageFaceEx** ..... 1  
..... 1  
..... 1  
..... 1  
..... 1

# BS2\_GetNormalizedImageFaceEx

[+ 2.8] FaceStation F2 WARP (가 ) , WARP  
 WARP 가,  
 WARP 가  
 WARP BS2FaceEx flag .

```
#include "BS_API.h"

int BS2_GetNormalizedImageFaceEx(void* context, uint32_t deviceId, const
uint8_t* unwarpedImage, uint32_t unwarpedImageLen, uint8_t* warpedImage,
uint32_t* warpedImageLen);
```

- [In] context : Context
- [In] deviceId :
- [In] unwarpedImage : WARP 가
- [In] unwarpedImageLen : unwarpedImage
- [Out] warpedImage : WARP
- [Out] warpedImageLen : warpedImage

BS\_SDK\_SUCCESS , 가

## C++

```
char* uidObj = NULL;
uint32_t numUID = ;
int sdkResult = BS2_GetUserList(context_, id, &uidObj, &numUID,
isAcceptableUID);
if (BS_SDK_SUCCESS != sdkResult)
{
  TRACE("BS2_GetUserList call failed: %d", sdkResult);
}
```

```
    return sdkResult;
}
// My Codes
//
// My Codes
BS2_ReleaseObject(uidObj);
```

C#

```
IntPtr outUidObjs = IntPtr.Zero;
UInt32 numUserIds = ;
API.IsAcceptableUserID cbIsAcceptableUserID = null; // we don't need to user
id filtering

IntPtr ptrDir = Marshal.StringToHGlobalAnsi(strDir);
BS2ErrorCode result = (BS2ErrorCode)API.BS2_GetUserList(sdkContext,
deviceID, out outUidObjs, out numUserIds, cbIsAcceptableUserID);
if (result != BS2ErrorCode.BS_SDK_SUCCESS)
{
    Console.WriteLine("Got error({0}).", result);
    return;
}
API.BS2_ReleaseObject(outUidObjs);
```

From:

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

Permanent link:

[https://kb.supremainc.com/kbtest/doku.php?id=ko:bs2\\_getnormalizedimagefaceex&rev=1652410864](https://kb.supremainc.com/kbtest/doku.php?id=ko:bs2_getnormalizedimagefaceex&rev=1652410864)

Last update: **2022/05/13 12:01**