

BS2_GetNormalizedImageFaceEx	1
.....	1
.....	1
.....	1
.....	1

Face API > [BS2_GetNormalizedImageFaceEx](#)

BS2_GetNormalizedImageFaceEx

[+ 2.8] Visual Face 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++

```
sdkResult = BS2_GetNormalizedImageFaceEx(context_, id, unwarpedImage.get(),
unwarpedImageLen, warpedImage.get(), &warpedImageLen);
if (BS_SDK_SUCCESS != sdkResult)
{
    TRACE("BS2_GetNormalizedImageFaceEx call failed: %d", sdkResult);
    return sdkResult;
}
```

```
}
```

C#

```
BS2ErrorCode result =
(BS2ErrorCode)API.BS2_GetNormalizedImageFaceEx(sdkContext, deviceID,
unwarpedImagePtr, unwarpedImageLen, warpedImagePtr, out warpedImageLen);
if (BS2ErrorCode.BS_SDK_SUCCESS == result)
{
    byte[] warpedBuffer = new byte[warpedImageLen];
    Array.Clear(warpedBuffer, , (int)warpedImageLen);
    Marshal.Copy(warpedImagePtr, warpedBuffer, , (int)warpedImageLen);

    Console.WriteLine("Enter the path and name of warped image file");
    Console.Write(">> ");
    string warpedPath = Console.ReadLine();
    if (warpedPath.Length == )
    {
        Console.WriteLine("The path and name can not be empty.");
        return;
    }

    File.WriteAllBytes(warpedPath, warpedBuffer);
}
else
{
    Console.WriteLine("Got error({0}).", result);
}

Marshal.FreeHGlobal(unwarpedImagePtr);
Marshal.FreeHGlobal(warpedImagePtr);
```

From:

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

Permanent link:

https://kb.supremainc.com/kbtest/doku.php?id=ko:bs2_getnormalizedimagefaceex

Last update: **2024/10/24 10:58**