

BS2_RemoveAuthGroup 1

..... 1

..... 1

..... 1

..... 1

..... 1

Face API > [BS2_RemoveAuthGroup](#)

BS2_RemoveAuthGroup

```
#include "BS_API.h"

int BS2_RemoveAuthGroup(void* context, uint32_t deviceId, uint32_t*
authGroupIds, uint32_t authGroupIdCount);
```

- [In] *context* : Context
- [In] *deviceId* :
- [In] *authGroupIds* : auth group id
- [In] *authGroupIdCount* : auth group id

BS_SDK_SUCCESS , 가

[BS2_GetAuthGroup](#)
[BS2_GetAllAuthGroup](#)
[BS2_SetAuthGroup](#)
[BS2_RemoveAllAuthGroup](#)

C#

```
Console.WriteLine("Enter the ID of the auth group which you want to remove:  
[ID_1,ID_2 ...]");  
Console.Write(">>>> ");  
char[] delimiterChars = { ' ', ',', '.', ':', '\\t' };  
string[] authGroupIDs = Console.ReadLine().Split(delimiterChars);  
List<UInt32> authGroupIDList = new List<UInt32>();
```

```
foreach (string authGroupID in authGroupIDs)
{
    if (authGroupID.Length > )
    {
        UInt32 item;
        if (UInt32.TryParse(authGroupID, out item))
        {
            authGroupIDList.Add(item);
        }
    }
}

if (authGroupIDList.Count > )
{
    IntPtr authGroupIDObj = Marshal.AllocHGlobal(4 * authGroupIDList.Count);
    IntPtr curAuthGroupIDObj = authGroupIDObj;
    foreach (UInt32 item in authGroupIDList)
    {
        Marshal.WriteInt32(curAuthGroupIDObj, (Int32)item);
        curAuthGroupIDObj = (IntPtr)((long)curAuthGroupIDObj + 4);
    }

    Console.WriteLine("Trying to remove auth groups from device.");
    result = (BS2ErrorCode)API.BS2_RemoveAuthGroup(sdkContext, deviceID,
authGroupIDObj, (UInt32)authGroupIDList.Count);

    Marshal.FreeHGlobal(authGroupIDObj);
}
else
{
    Console.WriteLine("Invalid parameter");
}
```

From:

<https://kb.supremainc.com/kbtest/> - **BioStar 2 Device SDK**

Permanent link:

https://kb.supremainc.com/kbtest/doku.php?id=ko:bs2_removeauthgroup

Last update: **2022/06/27 15:40**