

Table of Contents

BS2_GetAllAuthOperatorLevelEx

Declaration

Parameter

Return Value

See Also

.....

.....

.....

.....

.....

1

1

1

2

2

[Device API](#) > [BS2_GetAllAuthOperatorLevelEx](#)

BS2_GetAllAuthOperatorLevelEx

[+ 2.6.3] Gets all device operators.

When using [BS2_SetAuthConfig](#), [BS2_GetAuthConfig](#), there is a limitation that you can set operators up to 10.

The new API below accommodates setting operators up to 1,000.

Caution

The existing [BS2AuthConfig](#) and operators are valid until you call the new API. However, if at any one time you call the newly added API below, the existing [BS2AuthConfig](#) setting will be ignored so you need to use only the new API and structure.

If you want to use the new API with the new structure, please move the operator list managed by existing [BS2AuthConfig](#) securely to the new structure [BS2AuthOperatorLevel](#) where can manage 1000 operators.

Declaration

```
#include "BS_API.h"
```

```
int BS2_GetAllAuthOperatorLevelEx(void* context, BS2_DEVICE_ID deviceId,  
BS2AuthOperatorLevel** operatorlevelObj, uint32_t* numOperatorlevel);
```

[See BS2AuthOperatorLevel Structure](#)

Parameter

- [In] *context* : Context
- [In] *deviceId* : Device ID
- [Out] *operatorlevelObj* : Pointer to store the operators from device
- [Out] *numOperatorlevel* : Number of operators from device

Note

The *operatorlevelObj* variable needs to return the memory to the system by the [BS2_ReleaseObject](#) function after being used.

Return Value

If successfully done, BS_SDK_SUCCESS will be returned.

If there is an error, the corresponding error code will be returned.

See Also

[BS2_GetAuthOperatorLevelEx](#)

[BS2_GetAllAuthOperatorLevelEx](#)

[BS2_SetAuthOperatorLevelEx](#)

[BS2_RemoveAuthOperatorLevelEx](#)

[BS2_RemoveAllAuthOperatorLevelEx](#)

From:

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

Permanent link:

https://kb.supremainc.com/kbtest/doku.php?id=en:bs2_getallauthoperatorlevelex&rev=1559010823

Last update: **2019/05/28 11:33**