

BS2_GetAvailableOsdpStandardDevice

.....

.....

.....

.....

.....

.....

(C++)

(C#)

1

1

1

1

1

1

2

[Slave Control API](#) > [BS2_GetAvailableOsdpStandardDevice](#)

BS2_GetAvailableOsdpStandardDevice

[+ 2.9.1] CoreStation40

가

가

OSDP

가

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

```
int BS2_GetAvailableOsdpStandardDevice(void* context, uint32_t deviceId,  
BS2OsdpStandardDeviceAvailable* osdpDevices);
```

[BS2OsdpStandardDeviceAvailable](#)

- [In] *context* : Context
- [In] *deviceId* :
- [Out] *osdpDevices* : OSDP

BS_SDK_SUCCESS

, 가

[BS2_AddOsdpStandardDevice](#)
[BS2_GetOsdpStandardDevice](#)
[BS2_GetAvailableOsdpStandardDevice](#)
[BS2_UpdateOsdpStandardDevice](#)
[BS2_RemoveOsdpStandardDevice](#)
[BS2_GetOsdpStandardDeviceCapability](#)
[BS2_SetOsdpStandardDeviceSecurityKey](#)

(C++)[sample_bs2_GetAvailableOsdpStandardDevice.cpp](#)

```
BS2osdpStandardDeviceAvailable osdpDevice = { , };

BS2_DEVICE_ID id = Utility::selectDeviceID(deviceList, false, false);
int sdkResult = BS2_GetAvailableOsdpStandardDevice(context_, id,
&osdpDevice);
if (BS_SDK_SUCCESS != sdkResult)
    printf("BS2_GetAvailableOsdpStandardDevice call failed: %d",
sdkResult);

return sdkResult;
```

(C#)

[sample_bs2_GetAvailableOsdpStandardDevice.cs](#)

```
BS2osdpStandardDeviceAvailable osdpDevices;

BS2ErrorCode result =
(BS2ErrorCode)API.BS2_GetAvailableOsdpStandardDevice(sdkContext,
deviceID, out osdpDevices);
if (result != BS2ErrorCode.BS_SDK_SUCCESS)
    Console.WriteLine("Got error({0}).", result);
else
    Console.WriteLine("Call success.");

return result;
```

From:

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

Permanent link:

https://kb.supremainc.com/kbtest/doku.php?id=ko:bs2_getavailableosdpstandarddevice

Last update: **2023/02/14 14:51**