

BS2_SetOsdpStandardDeviceSecurityKey 1

..... 1

..... 1

..... 1

..... 1

(C++) 2

(C#) 2

BS2_SetOsdpStandardDeviceSecurityKey

[+ 2.9.1] CoreStation40 OSDP

가 ,

가 ,

, OSDP , 가

BS_SDK_ERROR_FAIL_MASTER_SET_KEY 가

OSDP 가

OSDP ,

OSDP

OSDP . (=NULL)

```
#include "BS_API.h"

int BS2_SetOsdpStandardDeviceSecurityKey(void* context, BS2_DEVICE_ID
masterOrSlaveId, const BS2OsdpStandardDeviceSecurityKey* key);
```

[BS2OsdpStandardDeviceSecurityKey](#)

- [In] *context* : Context
- [In] *masterOrSlaveId* :
- [In] *key* :

BS_SDK_SUCCESS , 가

[BS2_AddOsdpStandardDevice](#)
[BS2_GetOsdpStandardDevice](#)

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

(C++)

[sample_bs2_setosdpstandarddevicesecuritykey.cpp](#)

```
int sdkResult = BS_SDK_SUCCESS;

bool useMaster = true;
BS2_DEVICE_ID id =
(BS2_DEVICE_ID)Utility::selectMasterOrSlaveID(deviceList, useMaster);
if (useMaster)
{
    BS2OsdpStandardDeviceSecurityKey key = { , };
    string keyInfo = Utility::getInput<string>("Please enter the OSDP
security key.");
    memcpy(key.key, keyInfo.c_str(), min(keyInfo.size(),
BS2_OSDP_STANDARD_KEY_SIZE));

    sdkResult = BS2_SetOsdpStandardDeviceSecurityKey(context_, id,
&key);
}
else
{
    sdkResult = BS2_SetOsdpStandardDeviceSecurityKey(context_, id,
NULL);
}

if (BS_SDK_SUCCESS != sdkResult)
    printf("BS2_SetOsdpStandardDeviceSecurityKey call failed: %d",
sdkResult);

return sdkResult;
```

(C#)

[sample_bs2_setosdpstandarddevicesecuritykey.cs](#)

```
UInt32 id = Util.GetInputMasterOrSlaveID(deviceID);
BS2ErrorCode result = BS2ErrorCode.BS_SDK_SUCCESS;

if (id == deviceID)    // set key to master
{
```

```
BS2OsdpStandardDeviceSecurityKey keyInfo =
Util.AllocateStructure<BS2OsdpStandardDeviceSecurityKey>();

Util.HighlightLine(">>>> Please enter the OSDP security key.",
"security key");
Console.Write(">>>> ");
string keyString = Console.ReadLine();
byte[] buff = Encoding.UTF8.GetBytes(keyString);

Array.Clear(keyInfo.key, ,
BS2Environment.BS2_OSDP_STANDARD_KEY_SIZE);
Array.Copy(buff, , keyInfo.key, , keyString.Length);

IntPtr ptrKey =
Marshal.AllocHGlobal(Marshal.SizeOf(typeof(BS2OsdpStandardDeviceSecurityKey)));
Marshal.StructureToPtr(keyInfo, ptrKey, false);

result =
(BS2ErrorCode)API.BS2_SetOsdpStandardDeviceSecurityKey(sdkContext, id,
ptrKey);

Marshal.FreeHGlobal(ptrKey);
}
else
{
result =
(BS2ErrorCode)API.BS2_SetOsdpStandardDeviceSecurityKey(sdkContext, id,
IntPtr.Zero);
}

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 2 Device SDK**

Permanent link:
https://kb.supremainc.com/kbtest/doku.php?id=ko:bs2_setosdpstandarddevicesecuritykey

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