Table of Contents

Slave Control API > BS2_GetAvailableOsdpStandardDevice

BS2_GetAvailableOsdpStandardDevice

[+ 2.9.1] `CoreStation40` Get additional connectable OSDP device information by specifying the master device identifier.

Declaration

```
#include "BS_API.h"

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

> See BS2OsdpStandardDeviceAvailable Structure

Parameter

- [In] *context* : Context
- [In] *deviceId* : Master device identifier
- [Out] *osdpDevices* : Pointer to store OSDP device information

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_AddOsdpStandardDevice
BS2_GetOsdpStandardDevice
BS2_GetAvailableOsdpStandardDevice
BS2_UpdateOsdpStandardDevice
BS2_RemoveOsdpStandardDevice
BS2_GetOsdpStandardDeviceCapability
BS2_SetOsdpStandardDeviceSecurityKey

Sample Code(C++)

sample_bs2_GetAvailableOsdpStandardDevice.cpp

```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;
```

Sample Code (C#)

sample_bs2_GetAvailableOsdpStandardDevice.cs

```csharp
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;
```