Table of Contents

SDK API > BS2_QueryDeviceLicense

BS2_QueryDeviceLicense

[+ 2.9.1] Get the device license activation/deactivation status of the device connected as master-slave.
This feature is available only on devices that support the device license activation feature, and the devices that support the feature are listed below.

Supported devices	Firmware
XS2-Finger	V1.2.0
XS2-Card	V1.2.0
BS3	V1.1.0

Declaration

```
#include "BS_API.h"

int BS2_QueryDeviceLicense(void* context, uint32_t deviceId,
BS2_LICENSE_TYPE licenseType, BS2LicenseResult** outResultObj, uint32_t*
outNumOfResult);
```

> See BS2LicenseResult Structure

Parameter

- [In] *context* : Context
- [In] *deviceId* : Device Identifier
- [In] *licenseType* : Device License Type

Value	Description
0x0000	None
0x0001	Visual QR

- [Out] *outResultObj* : Pointer to structure to receive device license activation result
- [Out] *outNumOfResult* : Number of device license activation result structures

NOTE
The outResultObj parameter must be used and then returned memory to the system via the BS2_ReleaseObject function.

Return Value

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

Sample Code(C++)

sample_bs2_querydevicelicense.cpp

```
int getDeviceLicense(void* context, BS2_DEVICE_ID id)
{
    DeviceControl dc(context);
    BS2LicenseBlob licenseBlob = { , };
    vector<BS2LicenseResult> licenseResult;
    int sdkResult = BS_SDK_SUCCESS;

    BS2_LICENSE_TYPE licenseType =
(BS2_LICENSE_TYPE)Utility::getInput<uint32_t>("Enter the license type.
(0: None, 1: Visual QR)");
    sdkResult = dc.queryDeviceLicense(id, licenseType, licenseResult);
    if (BS_SDK_SUCCESS == sdkResult)
        DeviceControl::print(licenseResult);

    return sdkResult;
}

int DeviceControl::queryDeviceLicense(BS2_DEVICE_ID id,
BS2_LICENSE_TYPE licenseType, vector<BS2LicenseResult>& licenseResult)
{
    BS2LicenseResult* result = NULL;
    uint32_t numOfResult = ;
    int sdkResult = BS2_QueryDeviceLicense(context_, id, licenseType,
&result, &numOfResult);
    if (BS_SDK_SUCCESS != sdkResult)
    {
        TRACE("BS2_QueryDeviceLicense call failed: %d", sdkResult);
        return sdkResult;
    }

    licenseResult.clear();
    for (uint32_t idx = ; idx < numOfResult; idx++)
    {
        licenseResult.push_back(result[idx]);
    }

    return sdkResult;
}
```

Sample Code(C#)

sample_bs2_querydevicelicense.cs

```
    Console.WriteLine("Trying to get a license");

    Console.WriteLine("Enter the license type. (0: None, 1: Visual
QR)");
    Console.Write(">>>> ");
    UInt16 licenseType =
Util.GetInput((UInt16)BS2LicenseType.VISUAL_QR_MASK);

    IntPtr resultObj = IntPtr.Zero;
    UInt32 numOfResult = ;

    BS2ErrorCode result =
(BS2ErrorCode)API.BS2_QueryDeviceLicense(sdkContext, deviceID,
licenseType, out resultObj, out numOfResult);

    if (BS2ErrorCode.BS_SDK_SUCCESS != result)
    {
        Console.WriteLine("Got error({0}).", result);
    }
    else
    {
        IntPtr curResult = resultObj;
        int resultSize = Marshal.SizeOf(typeof(BS2LicenseResult));
        for (UInt32 idx = ; idx < numOfResult; idx++)
        {
            BS2LicenseResult item =
(BS2LicenseResult)Marshal.PtrToStructure(curResult,
typeof(BS2LicenseResult));
            print(item, idx);
            curResult += resultSize;
        }

        API.BS2_ReleaseObject(resultObj);
    }
```

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

Permanent link:
https://kb.supremainc.com/kbtest/doku.php?id=en:bs2_querydevicelicense

Last update: **2023/03/02 16:11**