

BS2_SetSSLHandler 1

..... 1

..... 1

..... 1

..... 1

BS2_SetSSLHandler

SSL

```
#include "BS_API.h"

int BS2_SetSSLHandler(void* context, PreferMethod ptrPreferMethod,
GetRootCaFilePath ptrGetRootCaFilePath, GetServerCaFilePath
ptrGetServerCaFilePath, GetServerPrivateKeyFilePath
ptrGetServerPrivateKeyFilePath, GetPassword ptrGetPassword, OnErrorOccured
ptrOnErrorOccured);
```

- [In] *context* : Context
- [In] *ptrPreferMethod* : SSL MethodMask
- [In] *ptrGetRootCaFilePath* : Root CA
- [In] *ptrGetServerCaFilePath* : CA
- [In] *ptrGetServerPrivateKeyFilePath* : PrivateKey
- [In] *ptrGetPassword* : Password
- [In] *ptrOnErrorOccured* : 가

BS_SDK_SUCCESS , 가

C++

```
int BS2Context::setSSLHandler()
{
    int sdkResult = BS2_SetSSLHandler(context_, BS2Context::OnPreferMethod,
BS2Context::OnGetRootCaFilePath, BS2Context::OnGetServerCaFilePath,
BS2Context::OnGetServerPrivateKeyFilePath, BS2Context::OnGetPassword,
BS2Context::onErrorOccured);
    if (BS_SDK_SUCCESS != sdkResult)
    {
```

```

        TRACE("BS2_SetSSLHandler call failed: %d", sdkResult);
        return sdkResult;
    }

    sdkResult = BS2_SetDeviceSSLEventListener(context_,
BS2Context::OnSendRootCA);
    if (BS_SDK_SUCCESS != sdkResult)
    {
        TRACE("BS2_SetDeviceSSLEventListener call failed: %d", sdkResult);
    }

    return sdkResult;
}

```

C#

```

if (Util.IsYes())
{
    private API.PreferMethod cbPreferMethod = null;
    private API.GetRootCaFilePath cbGetRootCaFilePath = null;
    private API.GetServerCaFilePath cbGetServerCaFilePath = null;
    private API.GetServerPrivateKeyFilePath cbGetServerPrivateKeyFilePath =
null;
    private API.GetPassword cbGetPassword = null;
    private API.OnErrorOccured cbOnErrorOccured = null;

    cbPreferMethod = new API.PreferMethod(PreferMethodHandle);
    cbGetRootCaFilePath = new
API.GetRootCaFilePath(GetRootCaFilePathHandle);
    cbGetServerCaFilePath = new
API.GetServerCaFilePath(GetServerCaFilePathHandle);
    cbGetServerPrivateKeyFilePath = new
API.GetServerPrivateKeyFilePath(GetServerPrivateKeyFilePathHandle);
    cbGetPassword = new API.GetPassword(GetPasswordHandle);
    cbOnErrorOccured = new API.OnErrorOccured(OnErrorOccuredHandle);
    //ServicePointManager.SecurityProtocol =
(SecurityProtocolType)SecurityProtocolType.Ssl3;

    BS2ErrorCode sdkResult = (BS2ErrorCode)API.BS2_SetSSLHandler(sdkContext,
cbPreferMethod, cbGetRootCaFilePath, cbGetServerCaFilePath,
cbGetServerPrivateKeyFilePath, cbGetPassword, null);
    if (sdkResult != BS2ErrorCode.BS_SDK_SUCCESS)
    {
        Console.WriteLine("BS2_SetSSLHandler failed with : {0}", sdkResult);
        API.BS2_ReleaseContext(sdkContext);
        sdkContext = IntPtr.Zero;
        return;
    }
    else
    {
        //bSsl = true;
    }
}

```

```
}  
  
}
```

From:

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

Permanent link:

https://kb.supremainc.com/kbtest/doku.php?id=ko:bs2_setsslhandler

Last update: **2022/06/29 08:35**