

BS2_WriteQRCode 1

..... 1

..... 1

..... 1

..... 1

BS2_WriteQRCode

[+ 2.8]	X-Station 2 CSN	QR	.		
QR	, ASCII	32(' ') 126('~')	,		32

```
#include "BS_API.h"
int BS2_WriteQRCode(const char* qrText, BS2CSNCard* card);
```

BS2CSNCard

- [In] *qrText* : QR
- [Out] *card* : QR 가 CSN

	BS_SDK_SUCCESS	,	
BS_SDK_ERROR_INVALID_PARAM	.		

C++

```
if (qrSupported)
{
    if (Utility::isYes("Would you like to register the QR code string to be used for authentication?"))
    {
        stringstream msg;
        msg << "Enter the ASCII QR code." << endl;
        msg << " [ASCII code consisting of values between 32 and 126].";
        string qrCode = Utility::getInput<string>(msg.str());
    }
}
```

```

    BS2CSNCard qrCard = { , };
    sdkResult = BS2_WriteQRCode(qrCode.c_str(), &qrCard);
    if (BS_SDK_SUCCESS != sdkResult)
    {
        TRACE("BS2_WriteQRCode call failed: %d", sdkResult);
    }
    else
    {
        size_t numOfRealloc = numOfCards + 1;
        BS2CSNCard* ptrNewCard = new BS2CSNCard[numOfRealloc];
        memset(ptrNewCard, 0x0, sizeof(BS2CSNCard) * numOfRealloc);

        if ( < numOfCards && *cardObjs)
        {
            memcpy(ptrNewCard, *cardObjs, sizeof(BS2CSNCard) *
numOfCards);

            delete[] * cardObjs;
            *cardObjs = NULL;
        }

        memcpy(ptrNewCard + numOfCards, &qrCard,
sizeof(BS2CSNCard));
        *cardObjs = ptrNewCard;
        numOfCards++;
    }
}
}

BS2_ReleaseObject(uidObj);

```

C#

```

    if (qrSupported)
    {
        Console.WriteLine("Would you like to register the QR code
string to be used for authentication? [y/n]");
        Console.Write(">>>> ");
        if (Util.IsYes())
        {
            Console.WriteLine("Enter the ASCII QR code.");
            Console.WriteLine(" [ASCII code consisting of values
between 32 and 126].");
            Console.Write(">>>> ");
            string qrCode = Console.ReadLine();

            IntPtr qrCodePtr = Marshal.StringToHGlobalAnsi(qrCode);
            BS2CSNCard qrCard =
Util.AllocateStructure<BS2CSNCard>();
            result = (BS2ErrorCode)API.BS2_WriteQRCode(qrCodePtr,
ref qrCard);

            if (BS2ErrorCode.BS_SDK_SUCCESS != result)

```

