

eNBSP - NBioBSP

NITGEN&COMPANY Biometric Service Provider SDK

Programmer's Manual ASP

SDK version 4.8x

© Copyright 2000-2011 NITGEN&COMPANY Co., Ltd.

ALL RIGHTS RESERVED

Serial Number:

Specifications subject to change without notice.

“NITGEN”, the NITGEN logo, “eNBSP”, “NBioBSP”, “NBioAPI”, “NITGEN Fingkey Mouse”, “eNDeSS”, “eNFolder”, and “eNFile” are trademarks of NITGEN&COMPANY Co., Ltd. All other brands or products may be trademarks or service marks of their respective owners.

목 차

제 1 장 ASP 프로그래밍	4
1.1 모듈 초기화	4
1.1.1 모듈 초기화	4
1.2 REGISTRATION	5
1.2.1 Object 설정을 위한 코드	5
1.2.2 지문 정보를 전달하기 위한 FORM	5
1.2.3 지문 등록을 위한 Javascript Code	5
1.2.4 지문 정보의 저장	7
1.3 VERIFICATION	7
1.3.1 Object 설정을 위한 코드	7
1.3.2 지문 정보를 전달하기 위한 FORM	7
1.3.3 지문 Capture 를 위한 Javascript Code	8
1.3.4 저장된 지문과의 비교	9

제1 장 ASP 프로그래밍

이 장에서는 NBioBSP COM 모듈을 이용한 ASP 프로그래밍에 대해 설명한다.

NBioBSP COM 모듈(NBioBSPCOM.dll)은 COM을 이용하는 언어를 지원하기 위한 목적으로 개발되었다.

NBioBSP COM 모듈은 NBioBSP에서 제공하는 모든 기능을 제공하지는 않는다. 지문 데이터는 웹 프로그래밍에서 사용할 수 있도록 텍스트 형태로 제공하며 Visual Basic 프로그래밍이나 Delphi 프로그래밍에서는 바이너리 형태의 지문 데이터와 텍스트 형태의 지문 데이터 포맷을 모두 지원한다.

이 장에서는 NBioBSP COM 모듈 사용법을 ASP를 예로 설명한다.

1.1 모듈 초기화

1.1.1 모듈 초기화

NBioBSP COM 모듈을 초기화 하는 방법은 CreateOleObject() 함수를 이용하여 초기화 하는 방법이 있다.

```
objNBioBSP : variant;           // Declaration variable for NBioBSP Object
...
objNBioBSP := CreateOleObject('NBioBSPCOM.NBioBSP');
...
```

NBioBSP COM 모듈은 COM 기반에서 작성되었기 때문에 Web 환경에서도 별도의 작업 없이 동작한다. 또한 Server Component 기능과 Client의 ActiveX 기능을 모두 포함하고 있기 때문에 Server 및 Client에서 동시에 사용할 수 있다.

1.2 Registration

사용자 등록을 위해서는 먼저 **Web Browser** 을 통하여 사용자의 지문 데이터를 등록 받아야 한다. 그리고 이 지문 데이터를 서버로 전송하여 **File** 또는 **DB** 에 저장하면 된다.

1.2.1 Object 설정을 위한 코드

Web Browser 에서 **NBioBSP COM** 모듈을 사용하기 위해서는 해당 **HTML** 페이지에 **Object** 로 설정하여야 한다.

```
<OBJECT classid="CLSID:F66B9251-67CA-4d78-90A3-28C2BFAE89BF"
        height=0 width=0
        id="objNBioBSP"
        name="objNBioBSP">
</OBJECT>
```

Javascript 코드에서 여기서 설정한 **Object** 의 이름을 이용하게 된다.

1.2.2 지문 정보를 전달하기 위한 FORM

아래 예는 지문 데이터를 서버로 전송하기 위한 **FORM** 이다. **Javascript** 에서 지문을 등록 받아 **FORM** 을 통하여 서버로 지문 데이터를 전송하게 된다. 이 **FORM** 이 **Submit** 될 때 지문 등록을 위한 **Javascript** 함수를 **Call** 하게 된다.

```
<form action='regist.asp' name='MainForm' method='post' OnSubmit='return regist();'>
<input type=hidden name='FIRTextData'>
User ID : <input type=text name=UserID size=20><p>
<input type=submit value=' Click here to register your fingerprint '>
</form>
```

여기서 설정한 **FORM** 의 버튼을 클릭하면 **Javascript** 의 **regist()** 함수를 불러 지문 등록 작업을 수행하게 된다. 이렇게 등록된 지문 등록 데이터는 **FIRTextData** 의 텍스트 항목으로 넘어오게 된다.

1.2.3 지문 등록을 위한 Javascript Code

웹 브라우저와 **NBioBSP COM** 모듈과의 통신을 위하여 **Javascript Code** 를 이용해야 한다. 여기서도 **Enroll** 메소드를 이용하게 된다. 위와 같은 흐름으로 함수를 구성하면 된다. **Javascript** 의 경우 대소문자 구분에 주의하도록 한다.

```
<script lang='javascript'>
function regist()
{
    var err, payload

    // Check ID is not NULL
    if ( document.MainForm.UserID.value == '' )
    {
        alert('Please enter user id !');
        return(false);
    }

    try // Exception handling
    {
```

```
// Open device. [AUTO_DETECT]
// You must open device before enroll.
DEVICE_FDU01 = 2;
DEVICE_FDU11 = 4;
DEVICE_AUTO_DETECT = 255;

var objDevice = document.objNBioBSP.Device;
var objExtraction = document.objNBioBSP.Extraction;

objDevice.Open(DEVICE_AUTO_DETECT);

err = objDevice.ErrorCode;          // Get error code

if ( err != 0 )                    // Device open failed
{
    alert('Device open failed !');
    return(false);
}

// Enroll user's fingerprint.
objExtraction.Enroll(payload);
err = objExtraction.ErrorCode;      // Get error code

if ( err != 0 )                    // Enroll failed
{
    alert('Registration failed ! Error Number : [' + err + ']');
    objDevice.Close(DEVICE_AUTO_DETECT);
    return(false);
}
else // Enroll success
{
    // Get text encoded FIR data from NBioBSP module.
    document.MainForm.FIRTextData.value = objExtraction.TextEncodeFIR;
    alert('Registration success !');
}

// Close device. [AUTO_DETECT]
objDevice.Close(DEVICE_AUTO_DETECT);

objExtraction = 0;
objDevice = 0;
}
catch(e)
{
    alert(e.message);
    return(false);
}

// Submit main form
document.MainForm.submit();
return(true);
}
</script>
```

1.2.4 지문 정보의 저장

ASP 코드를 이용하여 지문 데이터를 가져와 이를 서버에 있는 파일 또는 DB 에 저장하면 된다. 이렇게 저장된 지문 데이터를 이용하여 인증을 수행하게 된다.

```
<%
UserID      = Request.Form("UserID")
FIRTextData = Request.Form("FIRTextData")

' Write UserID and FIRTextData to File or DB.
%>
```

1.3 Verification

인증을 위한 지문을 **Capture** 받아 서버로 전송하여 서버에 저장된 지문 데이터와 비교하여 인증 작업을 수행하게 된다.

1.3.1 Object 설정을 위한 코드

Web Browser 에서 NBioBSP COM 모듈을 사용하기 위해서는 해당 HTML 페이지에 Object 로 설정하여야 한다.

```
<OBJECT classid="CLSID:F66B9251-67CA-4d78-90A3-28C2BFAE89BF"
        height=0 width=0
        id="objNBioBSP"
        name="objNBioBSP">
</OBJECT>
```

Javascript 코드에서 여기서 설정한 Object 의 이름을 이용하게 된다.

1.3.2 지문 정보를 전달하기 위한 FORM

아래 예는 Capture 된 지문 데이터를 서버로 전송하기 위한 FORM 이다. Javascript 에서 지문을 Capture 받아 여기 FORM 을 통하여 서버로 지문 데이터를 전송하게 된다. 이 FORM 이 Submit 될 때 지문 Capture 를 위한 Javascript 함수를 Call 하게 된다.

```
<form action='verify.asp' name='MainForm' method='post' OnSubmit='return capture();'>
<input type='hidden' name='FIRTextData'>
User ID : <input type='text' name=UserID size=20><p>
<input type='submit' value=' Click here to verification with your fingerprint '>
</form>
```

여기서 설정한 FORM 의 버튼을 클릭하면 Javascript 의 capture() 함수를 불러 지문 Capture 작업을 수행하게 된다. 이렇게 Capture 된 지문 데이터는 FIRTextData 의 텍스트 항목으로 넘어오게 된다.

1.3.3 지문 Capture 를 위한 Javascript Code

Javascript 의 경우 대소문자 구분에 주의하도록 한다.

```
<script lang='javascript'>

function capture()
{
    var err

    // Check ID is not NULL
    if ( document.MainForm.UserID.value == '' )
    {
        alert('Please enter user id !');
        return(false);
    }

    try // Exception handling
    {
        // Open device. [AUTO_DETECT]
        // You must open device before capture.
        DEVICE_FDU01 = 2;
        DEVICE_FDU11 = 4;
        DEVICE_AUTO_DETECT = 255;

        var objDevice = document.objNBioBSP.Device;
        var objExtraction = document.objNBioBSP.Extraction;

        objDevice.Open(DEVICE_AUTO_DETECT);
        err = objDevice.ErrorCode;          // Get error code

        if ( err != 0 )                    // Device open failed
        {
            alert('Device open failed !');
            return(false);
        }

        // Enroll user's fingerprint.
        objExtraction.Capture();
        err = objExtraction.ErrorCode;      // Get error code

        if ( err != 0 )                    // Enroll failed
        {
            alert('Capture failed ! Error Number : [' + err + ']');
            objDevice.Close(DEVICE_AUTO_DETECT);
            return(false);
        }
        else // Capture success
        {
            // Get text encoded FIR data from NBioBSP module.
            document.MainForm.FIRTextData.value = objExtraction.TextEncodeFIR;
            alert('Capture success !');
        }
    }
}
```



```

        // Close device. [AUTO_DETECT]
        objDevice.Close (DEVICE_AUTO_DETECT);

        objExtraction = 0;
        objDevice = 0;
    } // end try
    catch(e)
    {
        alert(e.message);
        return(false);
    }

    // Submit main form
    document.MainForm.submit();
    return(true);
}
</script>

```

1.3.4 저장된 지문과의 비교

먼저 등록된 지문 데이터를 읽어와 **Capture** 된 지문 데이터를 **VerifyMatch** 메소드를 이용하여 비교한다. 비교의 결과는 **MatchingResult** 속성에서 가져올 수 있다.

```

<%
    ' Read FIR data from file or DB.

    Set objNBioBSP = Server.CreateObject("NBioBSPCOM.NBioBSP")
    Set objMatching = objNBioBSP.Matching

    ' Verify Match
    ' FIRTextData is Caputured FIR
    ' fFIRTextData is FIR from file
    Call objMatching.VerifyMatch(CStr(FIRTextData), CStr(fFIRTextData))

    if objMatching.MatchingResult = 0 then
        ' Verification failed !!!
    else
        ' Verification success !!!
    end if

    ' Release NBioBSP object
    Set objMatching = nothing
    Set objNBioBSP = nothing
%>

```