

# MSDN アカデミックアライアンス ELMS 校内認証統合

山形大学工学部電気電子工学科 奥山澄雄 著\*

2008/12/10 ver.1.00

## 1 はじめに

2008年9月からMSDNアカデミックアライアンス(MSDN AA)を利用するためには利用者をHosted ELMSサイトへ登録することが義務化された。<sup>1)</sup> 利用者登録の方法の一つに、「校内の認証を統合」があり、利用者の認証を学内のサイトで行う方法がサポートされている。<sup>2)</sup> しかしながら「校内の認証の統合」についての[2]のドキュメントでの解説は手短でなかなか難しいものである。別に英語のドキュメント<sup>3)</sup>もあり、こちらでは実際のコードが例示されており、これを参考にするこゝとで実際のサイト構築を行うことができる。

本ドキュメントでは、実際に校内の認証の統合を行った手順・コードを紹介する。

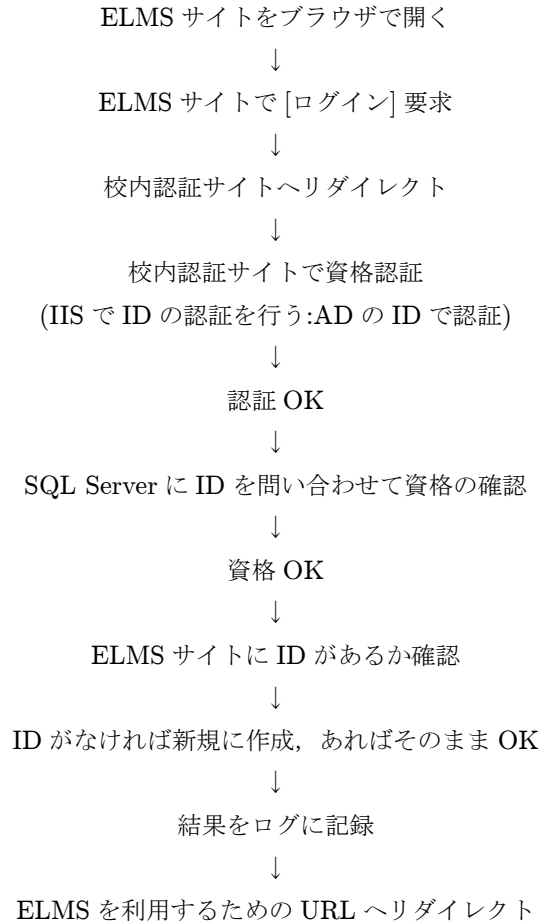
## 2 システム

### 2.1 用いた機材等

用いた機材等は下記のようなものになる。

- アクティブディレクトリ (AD)(ad.yamagata-u.ac.jp)  
Windows Server 2003
- WEBサーバー (web.ad.yamagata-u.ac.jp)  
Windows Server 2003 の IIS6.0. SSL(https) を有効化し、統合認証で認証される。認証の受け側のアドレスを <https://web.ad.yamagata-u.ac.jp/msdnaa/loginscript.aspx> とする。上記の AD のメンバサーバー。IP アドレスを 133.24.00.00 とする。
- データベース (db.ad.yamagata-u.ac.jp)  
Microsoft SQL Server 2005 上記の AD のメンバサーバー。
- 開発ツール  
Visual Studio 2005 の Visual Basic でコーディングを行った。 .NET Framework 2.0 をベースとする。

### 2.2 校内の認証の統合の流れ



## 3 設定方法

### 3.1 ELMS サイトの設定

#### 3.1.1 ELMS サイトの情報

e-academy から「MSDN アカデミック アライアンス: e-academy 使用許諾管理システム (ELMS)」といった表題のメールで、契約している MSDN AA 用の ELMS サイトの URL とユーザーネーム、パスワードが通知されている。たとえばこんな感じ。

- 所属学校・学科 (研究室) の ELMS の URL: [http://msdn51.e-academy.com/yamagatau\\_msdnaa](http://msdn51.e-academy.com/yamagatau_msdnaa)
- あなたのユーザーネーム: sumio@ieee.org
- あなたのパスワード: pppp1234

\* sumio@ieee.org

ELMS の URL のうち `yamagatau_msdnaa` の部分が ELMS サイトのユニーク文字列になるので重要である。

### 3.2 ELMS 側のパラメータの設定

通知された ELMS の URL にアクセスし、通知された管理 ID で [ログイン] を行う。<sup>\*1</sup>ログインできたらユーザー管理→統合された校内認証のリンクを選ぶと次のような画面になる。



#### 1. 校内認証

**テストモード:** ELMS サイト画面左上の [ログイン] リンクを押すと校内認証サイトへ飛ばされるようになる。校内認証で運用する場合はこちらでつかう。

**有効:** ドキュメントでは正式運用の際はこちらにする、と書いてあるが、設定をミスすると ELMS ログインによる管理ができなくなってしまうので要注意。

#### 2. 校内認証 URL

校内認証で認証を行う入り口ページの URL. たとえば `https://web.ad.yamagata-u.ac.jp/msdnaa/loginscript.aspx`

#### 3. 照合対象の学科

認証完了後に再度 ELMS サイトに戻るときに必要なパラメータ. 互換性を考えると英字で書いておくほうがいいかもしれない. たとえば `DenkiGakubu`

#### 4. おおよその学生/学科 (研究室) の数

在籍する学生+教職員の概数

#### 5. 構内用 CGI サーバーの IP

認証の可否を ELMS ホスト WEB サイトに送信するサーバーの IP アドレス. 通常は `web.ad.yamagata-u.ac.jp` の IP アドレスと同じになるが、校内の認証システムに柔軟に対応できるようにするため別途 IP アドレスを設定するようになっている様子. たとえば `133.24.00.00`

<sup>\*1</sup> 校内認証を「テストモード」にしたあとは下にある「ELMS ログイン」のリンクから、通知されたユーザーネーム、パスワードでログインを行う。校内認証を「有効」にするとこのリンクが消えてしまう。

#### 6. ELMS CGI コネクタ

ELMS サイトにユーザー ID があるかどうかをチェックするための URL. ELMS サイトにログインするにはこの URL に接続して、ユーザー ID の有無を確認し、その上で利用を行わなければならない。たとえば `https://msdn51.e-academy.com/yamagatau_msdnaa/index.cfm?loc=loginscript.aspx`

これ以下は実際のログインの際に ELMS サイトと校内認証サイトの間でやり取りされるパラメータになる。

##### 1. ユーザー名

たとえば `uid=sumio` といった形で、校内認証サイトから ELMS サイトへ送られる。学校側で決定してよい。

##### 2. 学生登録状況

たとえば `groups=all` といった形で、校内認証サイトから ELMS サイトへ送られる。学校側で決定し、あらかじめ ELMS サイトに設定しておく必要がある。ELMS サイトに管理者としてログインし、ユーザー管理→新しい登録単位の追加 から設定できる。

##### 3. ポータルに戻るリンク先の指定

たとえば `return_url=https://msdn51.e-academy.com/elms/Security/IntegratedLogin.aspx?campus=yamagatau_msdnaa` といった形で、ELMS サイトから校内認証サイトへ送られる。校内認証が成功した後、ELMS サイトを実際に利用するためにユーザーをリダイレクトする先になる。

##### 4. セキュリティトークン

たとえば `token=123456789` といった形で、ELMS サイトから校内認証サイトへ送られる。ログイン作業の識別を行うもので、一連の作業で必須になる。

##### 5. 学科

たとえば `department=DenkiGakubu` といった形で、ELMS サイトから校内認証サイトへ送られる。上記の照合対象の学科で設定した文字列と同一でなければならない。

##### 6. 名

`fname` : 必須ではないのでこの文書では使っていない

##### 7. 姓

`lname` : 必須ではないのでこの文書では使っていない

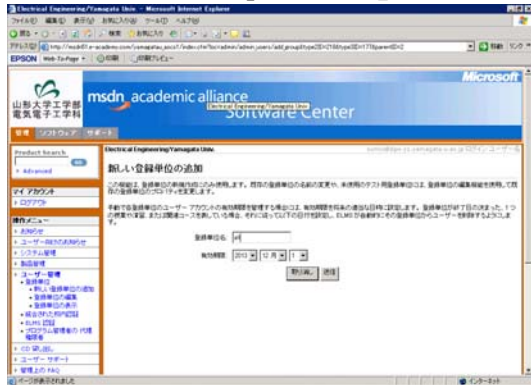
##### 8. email

`email` : 必須ではないのでこの文書では使っていない

ない

### 3.2.1 ELMS サイトの登録単位の設定

上記の ELMS の設定に「学生登録状況」の項目があるが、これはあらかじめ ELMS サイトに設定しておかなければならない。ELMS サイトに管理者としてログインし、ユーザー管理→新しい登録単位の追加 から設定できる。この画面で「all」グループをつくっておく。



## 4 認証

### 4.1 おおまかな流れ

学生の認証をどうやるかはいろいろなやり方があるが、ここでやったやり方は以下のような方法である。

1. ID は大学のコンピュータセンター (学術情報基盤センター) で発行したものをうい、この ID で認証を行う。この ID は大学の ad ドメインで管理されている。
2. 実際の認証は ad ドメインのメンバサーバーである、web.ad.yamagata-u.ac.jp 上の IIS で行われる。セキュリティのため通信は SSL で暗号化されている。IIS の設定で、SSL は必須にし、統合認証とする。
3. IIS での認証が通れば loginscript.aspx が動作し、資格認証のためバックエンドのデータベースへ接続される。バックエンドデータベースは ad ドメインのメンバサーバーである SQL Server 2005 で動作している。ユーザーが当該学科のユーザーであるかどうかを保持したテーブル tblMSDNAUser を参照して利用資格の有無を確認する。

### 4.2 データベース構造

セキュリティのため 2 つのデータベースに分離している。

#### ● LEVEL1

サイト情報: tblMSDNAASite

ユーザー情報: tblMSDNAUser

一般ユーザーは SELECT のみ。

#### ● LEVEL0

ログ: tblMSDNAALog

一般ユーザーは INSERT のみ。

### 4.3 テーブルの設計

#### 4.3.1 tblMSDNAASite

- nSiteID:プライマリーキー
- szDepartmentReal:学科の名前 (例:山形大学工学部電気電子工学科)
- szDepartmentElms:ELMS サイトに設定した Department (例:DenkiGakubu)
- szElmsCgiConnector:ELMS サイトに表示された ELMS CGI コネクタ. (例:https://msdn51.e-academy.com/yamagatau\_msdnaa/index.cfm?loc=login/cab\_cgi)
- szUniqString:サイトを識別するユニークな文字列 (例:yamagatau\_msdnaa)
- bDeleted:レコードの削除フラグ

#### 4.3.2 tblMSDNAUser

- nUserID:プライマリーキー
- szUserName:AD に登録している ID. この ID が ELMS サイト上にも作成される形になる (例:sumio)
- szStudentNumber:AD に登録されている学生番号 (例:081230123)
- szUserNameReal:学生の氏名 (例:奥山澄雄 7)
- nSiteID:このユーザーが所属する学科の nSiteID (例:1)
- szDepartmentElms:このユーザーが所属する学科の Department (例:DenkiGakubu)
- szGroup:このユーザーが所属するグループ (ELMS サイトに設定したグループ) (例:all)
- bValid:ID の有効/無効のフラグ
- szDepartmentAD:Active Directory にある所属学科名 (例:電気電子工学科)
- dRegistered:登録日時 (例: 2008/11/11 11:11:11)
- dUpdated:更新日時 (例: 2008/12/3 1:23:45)
- bDeleted:レコードの削除フラグ

#### 4.3.3 tblMSDNAALog

- nLogID:プライマリーキー
- szLogonUser:AD で認証されたユーザーの ID (例:sumio)
- nUserID:tblMSDNAUser 上の nUserID (例:1)
- szRemoteAddress:要求元の IP アドレス (例:111.22.33.44)

- dRegistered:登録日時 (例: 2008/11/11 11:11:11)
- nSiteID:このユーザーが所属する学科の nSiteID(例:1)
- bError:ログイン時のエラーの有無 (例:False)
- bIDExists:ユーザー ID の ELMS サイト上での有無 (例:True)
- bIDCreate:ユーザー ID の ELMS サイト上で新規作成されたかどうか (例:False)
- nSiteID:所属するサイトの nSiteID(例:1)
- szDepartmentReal:所属する学科の名前 (例:山形大学工学部電気電子工学科)
- bDeleted:レコードの削除フラグ

3.9」, <http://support.e-academy.com/admin/MsdnaaSupportDocuments.cfm>

## 5 プログラム

### 5.1 ログインのコード

ELMS サイトからのログイン要求を処理する。コードは `loginscript.aspx.vb` となる。SQL Server のテーブルを昔ながらの方法で引いている。IIS による統合ログインを行い、ID の偽装を行うため、`web.config` に `<identityimpersonate="true"/>` が必要である。

### 5.2 AD を検索し初期登録を行うコード

AD を検索し、条件にあったレコードを `tblMSDNAAUser` に登録または更新する。コードは `register.aspx.vb` となる。AD を検索するために `System.DirectoryServices` を用いる。`web.config` に設定が必要である。AD を検索し、条件にあったレコードを `tblMSDNAAUser` に登録または更新する。

## 6 おわりに

ここで紹介したコードは、「とにかく動くこと」を優先して書いたもので、ロジックや、使っているクラスライブラリやテーブルの設計など見直すべきところが多数あるので、自由に改変してお使いいただけたらと思う。

## 7 参考文献

1. 「Hosted ELMS を利用した学生 (アカデミックユーザー) 使用許諾の有効化」  
[http://www.microsoft.com/japan/academic/elms/easy\\_guide/u\\_reg/default.aspx](http://www.microsoft.com/japan/academic/elms/easy_guide/u_reg/default.aspx)
2. 「MSDNAA Hosted ELMS ソフトウェアソリューションを有効にする」, <http://www.microsoft.com/japan/academic/elms/manual/default.aspx>
3. 「ELMS-MSDNAA Integrated User Verification Customer Implementation Guide Version:

## 8 コード

### loginscript.aspx.vb

```
'IDの偽装を行うために web.config に<identity impersonate="true"/>が必要
Imports System.Data.SqlClient

Partial Public Class loginscript
    Inherits System.Web.UI.Page

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        Dim bError As Boolean = False 'エラーを検出した場合に True にする
        Label1.Text = "" '通常のメッセージは Label1 に書き込む
        LoginError.Visible = False 'デバッグ用の情報は LoginError に書き込む
        LoginError.Text = "エラー発生時には以下の情報を管理者あてお知らせください<br>"

        Dim token As String = Request.QueryString("token") '認証に必要なトークン:5分間だけ有効
        If token = "" Then token = "none" 'tokenが無いとエラーするので回避処理
        Dim return_url As String = Request.QueryString("return_url") '認証された後に飛ばす先の URL
        If return_url = "" Then return_url = "none" 'return_urlが無いとエラーするので回避処理
        Dim logon_user As String = Request.ServerVariables("LOGON_USER") '認証ユーザーの ID(たとえば username)
        Dim szServerName As String = Request.ServerVariables("SERVER_NAME") 'この.aspxが動作しているサーバの名前
        Dim szRemoteAddress As String = Request.ServerVariables("REMOTE_ADDR") 'リモートのアドレス
        LoginError.Text = LoginError.Text & "=====<br>"
        LoginError.Text = LoginError.Text & "time:" & DateTime.Now.ToString() & "<br>"
        LoginError.Text = LoginError.Text & "token:" & token & "<br>"
        LoginError.Text = LoginError.Text & "return_url:" & return_url & "<br>"
        LoginError.Text = LoginError.Text & "logon_user:" & logon_user & "<br>"
        LoginError.Text = LoginError.Text & "szServerName:" & szServerName & "<br>"
        LoginError.Text = LoginError.Text & "szRemoteAddress:" & szRemoteAddress & "<br>"

        '.aspx が web.ad.yamagata-u.ac.jp で動作しているときのみすべての動作を行う→セキュリティのため
        'デバッグは develop.ad.yamagata-u.ac.jp で行う
        If (szServerName Like "web.ad.yamagata-u.ac.jp") Or (szServerName Like "develop.ad.yamagata-u.ac.jp") Then
            LoginError.Text = LoginError.Text & "server check:OK<br>"

            'uidを規格化する username, ad\username, username@ad.yamagata-u.ac.jp とかでログイン可能のため統一する
            Dim uid As String = ""
            If logon_user Like "*ad\" Then 'ad\username とかの形式の場合
                uid = logon_user.Substring(logon_user.IndexOf("\") + 1)
            ElseIf logon_user Like "*@ad\" Then 'username@ad.yamagata-u.ac.jp とかの場合
                uid = logon_user.Substring(0, logon_user.IndexOf("@"))
            Else
                uid = logon_user 'これが通常
            End If
            LoginError.Text = LoginError.Text & "uid:" & uid & "<br>"

            'LEVEL1 DBへ接続 System.Data.SqlClient を用いている。
            '他に System.Data.OleDb や System.Data.Odbc でも可能()
            Dim connStr As String
            Dim conn As SqlConnection
            Dim sqlStr As String
            Dim cmd As SqlCommand
            Dim dr As SqlDataReader
            connStr = "Server=db.ad.yamagata-u.ac.jp;" & _
                & "Integrated Security=SSPI;" & _
                & "Trusted_Connection=Yes;" & _
                & "Database=LEVEL1;"
            conn = New SqlConnection(connStr)

            'サイトの情報との照合
            conn.Open()
            'tblMSDNAASiteのすべての有効なレコードをセレクト
            sqlStr = "SELECT * FROM tblMSDNAASite WHERE bDeleted='FALSE'"
            cmd = New SqlCommand(sqlStr, conn)
            dr = cmd.ExecuteReader()
            Dim bDepartmentMatch As Boolean = False 'クエリの学科が tblMSDNAASite に存在すれば True になる
            Dim nSiteID As Integer = 0 'tblMSDNAASiteでのサイトのID
            Dim szDepartmentReal As String = "" '学科名
            Dim elms_cgi_connector As String = "" 'IDの状態をチェックするためのURL:ELMS管理IDで表示される
            Dim department As String = "" 'ELMSサイトに設定したdepartmentの値=tblMSDNAASiteに保存してある

            While (dr.Read())
                'DBにサイト固有の文字列 yamagatau_msdnaa を szUniqString として格納してある
                If return_url Like ("*" & dr("szUniqString") & "*") Then
                    bDepartmentMatch = True
                    nSiteID = dr("nSiteID")
                    szDepartmentReal = dr("szDepartmentReal")
                    elms_cgi_connector = dr("szElmsCgiConnector")
                    department = dr("szDepartmentElms")
                    LoginError.Text = LoginError.Text & "site record: exists<br>"
                    LoginError.Text = LoginError.Text & "nSiteID:" & nSiteID.ToString() & "<br>"
                    LoginError.Text = LoginError.Text & "szDepartmentReal:" & szDepartmentReal & "<br>"
                    LoginError.Text = LoginError.Text & "elms_cgi_connector:" & elms_cgi_connector & "<br>"
                    LoginError.Text = LoginError.Text & "department:" & department & "<br>"
                End If
            End While
            If Not (bDepartmentMatch) Then bError = True '学科が一致しなければエラー
            dr.Close()
            conn.Close()

            'ユーザー情報との照合
            '(ユーザー名)and(nSiteID)のレコードがテーブルにあるかどうかをチェック
            conn.Open()
            sqlStr = "SELECT * FROM tblMSDNAUser " & _
                & "WHERE szUserName='" & uid & "' " & _
                & "AND nSiteID='" & nSiteID.ToString() & "' " & _
                & "AND bDeleted='FALSE'"
            cmd = New SqlCommand(sqlStr, conn)
            dr = cmd.ExecuteReader()
            Dim nUserID As Integer = 0 'tblMSDNAUserのプライマリーキー
            Dim bValid As Boolean = False 'IDが有効かどうか
            'ユーザーが所属するgroupの名前:tblMSDNAUserに設定してある。
            '別途ELMSサイトにも設定しなければならない()
            Dim groups As String = ""
            'tblMSDNAUserテーブルにIDがあるかどうかのチェック
            If (dr.Read()) Then 'IDがある場合
                nUserID = dr("nUserID")
                bValid = dr("bValid")
                groups = dr("szGroup")
            Else 'IDがない場合
                nUserID = 0
                bValid = False
                groups = ""
            End If
            LoginError.Text = LoginError.Text & "nUserID:" & nUserID.ToString() & "<br>"
            LoginError.Text = LoginError.Text & "bValid:" & bValid.ToString() & "<br>"
            LoginError.Text = LoginError.Text & "groups:" & groups & "<br>"
        End If
    End Sub
End Class
```

```

If Not (bValid) Then bError = True ' 有効な ID が確認できなければエラー
dr.Close()
conn.Close()

'DB のレコードから groups の値が確定したので下記の URL を作成できる
'ID の状態をチェックするための URL
Dim URL_ID_CHECK As String = elms.cgi_connector + "&token=" + token + "&uid=" + uid + "&groups=" + groups + "&department=" + department
' 認証ができた場合に飛ばす先の URL
Dim URL_REDIRECT As String = return_url + "&token=" + token + "&uid=" + uid + "&groups=" + groups + "&department=" + department

' 当該 ELMS サイトに ID があるかどうかをチェックする
Dim HttpWReq As Object 'http web request を行うオブジェクト
Dim HttpWResp As Object 'request の結果を格納するオブジェクト
Dim bIDExists As Boolean = False 'ELMS サイトに ID が存在すれば True
Dim bIDCreate As Boolean = False 'ELMS サイトに新規に ID が作成されれば True
' ここまでのチェックでエラーおよび明らかな不具合が無いときのみ
'ELMS サイトで ID を確認する
If (Not bError) And (bDepartmentMatch) And (bValid) And (return_url <> "") And (token <> "") Then
    'ELMS サイトに ID があるかどうかのチェック
    HttpWReq = System.Net.WebRequest.Create(URL_ID_CHECK)
    HttpWReq.Timeout = 50000
    HttpWResp = HttpWReq.GetResponse()

    ' レスポンスが正しく返ってくるまで待つ
    If (HttpWResp.StatusCode = System.Net.HttpStatusCode.OK) Then
        Dim temp1 As String = ""
        Dim st As IO.Stream = HttpWResp.GetResponseStream()
        Dim webstream As New IO.StreamReader(st, System.Text.Encoding.ASCII())

        ' レスポンスの文字列をすべて格納
        Do While webstream.Peek() >= 0
            temp1 += webstream.ReadLine()
        Loop
        webstream.Close()
        LoginError.Text = LoginError.Text & "Response of ID Check:" & temp1 & "<br>"

        'ELMS サイトからのレスポンスにより ID の有無がわかる
        If temp1 Like "*0 Account *" Then 'ELMS に ID が存在 | 作成された場合=成功
            If temp1 Like "*0 Account updated*" Then 'ELMS に既に有効なアカウントが存在する場合
                bIDExists = True
                LoginError.Text = LoginError.Text & "There exists valid ID on the ELMS site.<br>"
            ElseIf temp1 Like "*0 Account created*" Then 'ELMS サイトでアカウントが作成された場合
                bIDCreate = True
                LoginError.Text = LoginError.Text & "Valid ID was created on the ELMS site.<br>"
            Else ' 何らかのエラーが発生した場合の処理
                bError = True
                LoginError.Text = LoginError.Text & "再試行が必要です:" & temp1 & "<br>"
            End If
        Else ' 何らかのエラーが発生した場合の処理
            bError = True
            LoginError.Text = LoginError.Text & "再試行が必要です:" & temp1 & "<br>"
        End If
    Else ' 何らかのエラーが発生した場合の処理
        bError = True
        LoginError.Text = LoginError.Text & "ELMS サイトからのレスポンスがありません" & "<br>"
    End If
Else ' 何らかのエラーが発生した場合の処理
    bError = True
    LoginError.Text = LoginError.Text & "必要な条件を満たしていません" & "<br>"
End If

' 結果をログに書き込むために LEVEL0 DB へ接続
connStr = "Server=db.ad.yamagata-u.ac.jp;" & _
    & "Integrated Security=SSPI;" & _
    & "Trusted_Connection=Yes;" & _
    & "Database=LEVEL0;"
conn = New SqlConnection(connStr)

conn.Open()
sqlStr = "INSERT INTO tblMSDNAALog " & _
    & "(szLogonUser, " & _
    & "nUserID, " & _
    & "szRemoteAddress, " & _
    & "nSiteID, " & _
    & "szDepartmentReal, " & _
    & "dRegistered, " & _
    & "bDeleted, " & _
    & "bError, " & _
    & "bIDExists, " & _
    & "bIDCreate) " & _
    & "VALUES " & _
    & "(" & logon_user & ", " & _
    & "" & nUserID.ToString() & ", " & _
    & "" & szRemoteAddress & ", " & _
    & "" & nSiteID.ToString() & ", " & _
    & "" & szDepartmentReal & ", " & _
    & "" & DateTime.Now.ToString() & ", " & _
    & "False", " & _
    & "" & bError.ToString() & ", " & _
    & "" & bIDExists.ToString() & ", " & _
    & "" & bIDCreate & ")"
cmd = New SqlCommand(sqlStr, conn)
dr = cmd.ExecuteReader()
conn.Close()

' 最終判定を行う
If Not (bDepartmentMatch) Then 'DB に当該 Department が見つからない
    Label1.Text = Label1.Text & "正しくない呼び出しです<br>"
    LoginError.Text = LoginError.Text & "正しくない呼び出しです<br>"
    Label1.Text = Label1.Text & "管理者に問い合わせてください<br>"
    LoginError.Visible = True
ElseIf Not (bValid) Then ' 有効な ID が確認できなかった場合
    Label1.Text = Label1.Text & "利用資格が確認できませんでした<br>"
    LoginError.Text = LoginError.Text & "利用資格が確認できませんでした<br>"
    Label1.Text = Label1.Text & "管理者に問い合わせてください<br>"
    LoginError.Visible = True
ElseIf (bError) Then ' 上記以外のエラーが発生した場合
    Label1.Text = Label1.Text & "エラーが発生しました"
    Label1.Text = Label1.Text & "管理者に問い合わせてください<br>"
    LoginError.Visible = True
Else ' エラーが起きなければリダイレクトして処理完了
    Response.Redirect(URL_REDIRECT)
End If
End If
End Sub
End Class

```

## loginscript.aspx

```
<%@ Page Language="vb" AutoEventWireup="false" CodeBehind="loginscript.aspx.vb" Inherits="amenity.loginscript" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>MSDN AA ELMS Integrated Login</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:Label ID="Label1" runat="server" Text="Label1"></asp:Label><br />
      <asp:Label ID="LoginError" runat="server" Text="LoginError"></asp:Label>&nbsp;</div>
    </form>
  </body>
</html>
```

## loginscript.aspx.designer.vb

```
'-----  
' <auto-generated>  
' このコードはツールによって生成されました。  
' ランタイム バージョン:2.0.50727.1433  
'  
' このファイルへの変更は、以下の状況下で不正な動作の原因になったり、  
' コードが再生成されるときに損失したりします。  
' </auto-generated>  
'-----  
  
Option Strict Off  
Option Explicit On  
  
'''<summary>  
'''loginscript クラス。  
'''</summary>  
'''<remarks>  
''' 自動生成されたクラス。  
'''</remarks>  
Partial Public Class loginscript  
  
    '''<summary>  
    '''form1 コントロール。  
    '''</summary>  
    '''<remarks>  
    ''' 自動生成されたフィールド。  
    ''' 変更するには、フィールドの宣言をデザイナ ファイルから分離コード ファイルに移動します。  
    '''</remarks>  
    Protected WithEvents form1 As Global.System.Web.UI.HtmlControls.HtmlForm  
  
    '''<summary>  
    '''Label1 コントロール。  
    '''</summary>  
    '''<remarks>  
    ''' 自動生成されたフィールド。  
    ''' 変更するには、フィールドの宣言をデザイナ ファイルから分離コード ファイルに移動します。  
    '''</remarks>  
    Protected WithEvents Label1 As Global.System.Web.UI.WebControls.Label  
  
    '''<summary>  
    '''LoginError コントロール。  
    '''</summary>  
    '''<remarks>  
    ''' 自動生成されたフィールド。  
    ''' 変更するには、フィールドの宣言をデザイナ ファイルから分離コード ファイルに移動します。  
    '''</remarks>  
    Protected WithEvents LoginError As Global.System.Web.UI.WebControls.Label  
End Class
```



## register.aspx.vb

```
'System.DirectoryServices を使うには web.config に設定が必要
Imports System.DirectoryServices
Imports System.Data
Imports System.data.SqlClient

Partial Public Class register
    Inherits System.Web.UI.Page

    Private Sub register_Init(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Init
        Label1.Text = ""
        LabelError.Text = "デバッグ情報<br>"
        ' チェックボタンを無効化
        ButtonCheck.Enabled = False
        ' 登録ボタンを無効化
        ButtonRegister.Enabled = False

        ' 作業環境を確認し、セキュアにする
        Dim logon_user As String = Request.ServerVariables("LOGON_USER") ' 認証ユーザーの ID(username)
        LabelError.Text = LabelError.Text & "logon_user:" & logon_user & "<br>"
        Dim szServerName As String = Request.ServerVariables("SERVER_NAME") ' この.aspx が動作しているサーバの名前
        LabelError.Text = LabelError.Text & "szServerName:" & szServerName & "<br>"
        Dim szRemoteAddress As String = Request.ServerVariables("REMOTE_ADDR") ' リモートのアドレス
        LabelError.Text = LabelError.Text & "szRemoteAddress:" & szRemoteAddress & "<br>"
        Dim bSecure As Boolean = False
        If (szServerName Like "web.ad.yamagata-u.ac.jp") Or (szServerName Like "localhost") Or (szServerName Like "develop.ad.yamagata-u.ac.jp") Then
            If (szRemoteAddress Like "133.24.*") Or (szRemoteAddress Like "127.0.0.1") Then
                If (logon_user.ToLower() Like "*sum*") Or (logon_user.ToLower() Like "*tom*") Then
                    bSecure = True
                End If
            End If
        End If
    End Sub

    ' 作業環境が確認されたときのみ処理を続ける
    If (bSecure) Then
        ' チェックボタンを有効化
        ButtonCheck.Enabled = True
        ' 登録ボタンを無効化
        ButtonRegister.Enabled = False
        ' グループ設定のドロップダウンリストの初期化
        DropDownListGroups.Items.Add("all")
        ' 有効 | 無効のドロップダウンリストの初期化
        Dim myListItem1 As ListItem = New ListItem
        With myListItem1
            myListItem1.Text = "有効 (True)"
            myListItem1.Value = "True"
        End With
        DropDownListValid.Items.Add(myListItem1)
        Dim myListItem2 As ListItem = New ListItem
        With myListItem2
            myListItem2.Text = "無効 (False)"
            myListItem2.Value = "False"
        End With
        DropDownListValid.Items.Add(myListItem2)

        'LEVEL1 DB へ接続 System.Data.SqlClient を用いている。他に System.Data.OleDb や System.Data.Odbc でも可能
        Dim connStr As String
        Dim conn As SqlConnection
        Dim sqlStr As String
        Dim cmd As SqlCommand
        Dim dr As SqlDataReader
        connStr = "Server=db.ad.yamagata-u.ac.jp;" & _
            & "Integrated Security=SSPI;" & _
            & "Trusted_Connection=Yes;" & _
            & "Database=LEVEL1;"
        conn = New SqlConnection(connStr)

        ' ドロップダウンリストに department 情報を設定する
        conn.Open()
        sqlStr = "SELECT * FROM tblMSDNAASite ORDER BY nSiteID"
        cmd = New SqlCommand(sqlStr, conn)
        dr = cmd.ExecuteReader()
        Dim nSiteID As Integer = 0
        Dim department As String = ""
        Dim departmentReal As String = ""
        While (dr.Read())
            Dim myListItem As ListItem = New ListItem
            nSiteID = dr("nSiteID")
            department = dr("szDepartmentElms")
            departmentReal = dr("szDepartmentReal")
            myListItem.Text = departmentReal & ":" & department
            myListItem.Value = nSiteID
            DropDownListDepartment.Items.Add(myListItem)
            'LabelError.Text = LabelError.Text & "department:" & department & "<br>"
        End While
        dr.Close()
        conn.Close()
    End If
End Sub

Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
End Sub

Protected Sub ButtonCheck_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles ButtonCheck.Click
    Dim register As Boolean = False
    DataHandling(register)
    ButtonRegister.Enabled = True
End Sub

Protected Sub ButtonRegister_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles ButtonRegister.Click
    Dim register As Boolean = True
    DataHandling(register)
    ButtonRegister.Enabled = False
End Sub

Private Sub DataHandling(ByVal register As Boolean)
    'DirectoryEntry オブジェクトを生成する
    Dim entry As New DirectoryEntry("LDAP://DC=ad,DC=yamagata-u,DC=ac,DC=jp")
    'DirectorySearcher オブジェクトを生成する
    Dim mySearcher As New DirectorySearcher(entry)

    'LEVEL1 DB へ接続 System.Data.SqlClient を用いている。他に System.Data.OleDb や System.Data.Odbc でも可能
    Dim connStr As String
    Dim conn As SqlConnection
    Dim sqlStr As String
    Dim cmd As SqlCommand
    Dim dr As SqlDataReader
    connStr = "Server=db.ad.yamagata-u.ac.jp;" & _
```

```

& "Integrated Security=SSPI;" _
& "Trusted_Connection=Yes;" _
& "Database=LEVEL1;"
conn = New SqlConnection(connStr)

'AD の検索キーワードを設定する
mySearcher.Filter = "((&(displayName=" & TextBox1.Text & ")(&(sAMAccountName=" & TextBox2.Text & ")))"
'FindOne メソッドで検索を実行し、ResEnt に入れる
'Dim ResEnt As SearchResult = mySearcher.FindOne()
'FindAll メソッドだと検索された結果がコレクションで返される
Dim myResEntCol As SearchResultCollection = mySearcher.FindAll() 'ここで AD の検索が実行されている
Dim myResEnt As SearchResult
Label1.Text = ""
For Each myResEnt In myResEntCol
    'Label1.Text = ResEnt.Path & "<br>"
    'Assign a property name to propKey.
    'プロパティを格納する propKey 変数の確保
    Dim propKey As String
    Dim nUserID As Int32
    Dim szStudentNumber As String = "" ' 学生番号
    Dim szUserName As String = "" ' ログイン ID
    Dim szUserNameReal As String = "" ' 本人の名前
    Dim nSiteID As Int32 = 0 'tblMSDNAASite の nSiteID
    Dim szDepartmentElms As String = "" 'ELMS の Department
    Dim szGroup As String = "" 'ELMS の学科
    Dim szValid As Boolean = True 'ELMS を利用する資格の有無
    Dim szDepartmentAD As String = "" 'AD に書き込んである所属の情報
    Dim bSNEExist As Boolean = False 'SirName (苗字) 情報が AD にあるかどうか
    Dim bGivenNameExist As Boolean = False 'GivenName (名前) 情報が AD にあるかどうか
    'szDepartmentElms = DropDownListDepartment.SelectedItem.Text.Substring(DropDownListDepartment.SelectedItem.Text.IndexOf(".") + 1)
    szDepartmentElms = DropDownListDepartment.SelectedItem.Text.Substring(DropDownListDepartment.SelectedItem.Text.IndexOf(".") + 1)
    nSiteID = DropDownListDepartment.SelectedItem.Text
    szGroup = DropDownListGroups.SelectedItem ' 選択された Group
    szValid = DropDownListValid.SelectedItem ' 選択された有効/無効
    'propKey にプロパティを順次割り当てて AD の情報を読み出す
    For Each propKey In myResEnt.Properties.PropertyNames
        Select Case propKey
            Case "displayName" 'AD では学生番号に相当する
                szStudentNumber = myResEnt.Properties("displayName")(0).ToString()
            Case "samaccountname" 'ad のログイン ID
                szUserName = myResEnt.Properties("samaccountname")(0).ToString()
            Case "sn" 'SirName フィールドの有無
                bSNEExist = True
            Case "givenname" 'GivenName フィールドの有無
                bGivenNameExist = True
            Case "physicaldeliveryofficename" 'AD にある所属の情報
                szDepartmentAD = myResEnt.Properties("physicaldeliveryofficename")(0).ToString()
            Case Else
                End Select
        '実際に AD を全部読み出す デバッグ用
        'Dim prop As [Object]
        'For Each prop In myResEnt.Properties(propKey)
        '    LabelError.Text = LabelError.Text & prop.ToString() & "| "
        'Next prop
    Next propKey
    'SirName および GivenName の有無で szUserNameReal を生成する
    If bSNEExist And bGivenNameExist Then
        szUserNameReal = myResEnt.Properties("sn")(0).ToString().Trim() & " " & myResEnt.Properties("givenname")(0).ToString().Trim()
    ElseIf bSNEExist And Not bGivenNameExist Then
        szUserNameReal = myResEnt.Properties("sn")(0).ToString().Trim()
    ElseIf Not bSNEExist And bGivenNameExist Then
        szUserNameReal = myResEnt.Properties("givenname")(0).ToString().Trim()
    ElseIf Not bSNEExist And Not bGivenNameExist Then
        szUserNameReal = ""
    End If

    'DB を読み出して ID の有無を確認する
    conn.Open()
    'szUserName と nSiteID で唯一のレコードが選ばれるコトになっている
    sqlStr = "SELECT * FROM tblMSDNAUser " _
        & "WHERE szUserName='" & szUserName & "' " _
        & "AND nSiteID='" & nSiteID.ToString() & "' " _
        & "AND szDepartmentElms='" & szDepartmentElms & "'"
    cmd = New SqlCommand(sqlStr, conn)
    dr = cmd.ExecuteReader()
    nUserID = 0
    If (dr.Read()) Then
        nUserID = dr("nUserID")
        Label1.Text = Label1.Text & "<font color=""red""><b>有</b></font>" & "|"
        Label1.Text = Label1.Text & szStudentNumber & "|"
        Label1.Text = Label1.Text & szUserName & "|"
        Label1.Text = Label1.Text & szUserNameReal & "|"
        Label1.Text = Label1.Text & szDepartmentElms & "|"
        Label1.Text = Label1.Text & szGroup & "|"
        If dr("bValid").ToString() = szValid Then
            Label1.Text = Label1.Text & szValid & "|"
        Else
            Label1.Text = Label1.Text & "<font color=""red""><b>" & szValid & "</b></font>" & "|"
        End If
        Label1.Text = Label1.Text & szDepartmentAD & "|"
        Label1.Text = Label1.Text & "<br>"
    Else
        nUserID = 0
        Label1.Text = Label1.Text & "無" & "|"
        Label1.Text = Label1.Text & szStudentNumber & "|"
        Label1.Text = Label1.Text & szUserName & "|"
        Label1.Text = Label1.Text & szUserNameReal & "|"
        Label1.Text = Label1.Text & szDepartmentElms & "|"
        Label1.Text = Label1.Text & szGroup & "|"
        Label1.Text = Label1.Text & szValid & "|"
        Label1.Text = Label1.Text & szDepartmentAD & "|"
        Label1.Text = Label1.Text & "<br>"
    End If
dr.Close()
If (registor) Then
    If (nUserID = 0) Then
        'レコードを追加する
        sqlStr = "INSERT INTO tblMSDNAUser " _
            & "(" _
            & "szStudentNumber," _
            & "szUserName," _
            & "szUserNameReal," _
            & "nSiteID," _
            & "szDepartmentElms," _
            & "szGroup," _
            & "bValid," _
            & "szDepartmentAD," _
            & "dRegistered," _
            & "dUpdated," _
            & "bDeleted" _

```

```

& ") VALUES (" _
& "'" & szStudentNumber & "'," _
& "'" & szUserName & "'," _
& "'" & szUserNameReal & "'," _
& "'" & nSiteID.ToString() & "'," _
& "'" & szDepartmentElms & "'," _
& "'" & szGroup & "'," _
& "'" & szValid & "'," _
& "'" & szDepartmentAD & "'," _
& "'" & DateTime.Now().ToString() & "'," _
& "'" & DateTime.Now().ToString() & "'," _
& "'" & False.ToString() & "'," _
& ")"
'LabelError.Text = LabelError.Text & sqlStr
cmd = New SqlCommand(sqlStr, conn)
dr = cmd.ExecuteReader()
dr.Close()
Else
'szUserID で示されるレコードを更新する
sqlStr = "UPDATE tblMSDNAAUser " _
& "SET " _
& "szStudentNumber=" & szStudentNumber & "'," _
& "szUserName=" & szUserName & "'," _
& "szUserNameReal=" & szUserNameReal & "'," _
& "nSiteID=" & nSiteID.ToString() & "'," _
& "szDepartmentElms=" & szDepartmentElms & "'," _
& "szGroup=" & szGroup & "'," _
& "bValid=" & szValid & "'," _
& "szDepartmentAD=" & szDepartmentAD & "'," _
& "dUpdated=" & DateTime.Now().ToString() & "'," _
& "bDeleted=" & False.ToString() & "'," _
& "WHERE nUserID=" & nUserID.ToString() & "'"
'LabelError.Text = LabelError.Text & sqlStr
cmd = New SqlCommand(sqlStr, conn)
dr = cmd.ExecuteReader()
dr.Close()
End If
Label1.Text = Label1.Text & "<font color=""red""><b>登録完了</b></font>" & "<br>"
End If
conn.Close()
Next myResEnt
End Sub
End Class

```

# register.aspx

```
<%@ Page Language="vb" AutoEventWireup="false" CodeBehind="register.aspx.vb" Inherits="amenity.register" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>AD → tblMSDNAAUser メンテナンス</title>
</head>
<body>
<form id="form1" runat="server">
<div>
Active Directory を検索し、ユーザーを tblMSDNAAUser に登録します。 <br />
(学生番号と ID の AND で検索されます)<br />
tblMSDNAAUser に書き込み権限のあるアカウントでログインしてください。 <br />
===== <br />
学生番号 (AD の displayname) : <asp:TextBox ID="TextBox1" runat="server">08123456</asp:TextBox>&nbsp;<br />
081* : 電気 2008 入学、0*1* : 電気 全員 <br />
ad での ID (AD の sAMAccountName) : <asp:TextBox ID="TextBox2" runat="server">*</asp:TextBox><br />
MSDN AA の Department : <asp:DropDownList ID="DropDownListDepartment" runat="server">
</asp:DropDownList><br />
MSDN AA の groups : <asp:DropDownList ID="DropDownListGroups" runat="server">
</asp:DropDownList><br />
有効化 | 無効化 (bValid) : <asp:DropDownList ID="DropDownListValid" runat="server">
</asp:DropDownList><br />
===== <br />
<asp:Button ID="ButtonCheck" runat="server" Text="チェック" />→チェックがうまくいったら下の [登録] ボタン<br />
===== <br />
<asp:Label ID="Label1" runat="server" Text="Label1"></asp:Label><br />
===== <br />
<asp:Button ID="ButtonRegister" runat="server" Text="登録" />→レコードが無いものは新規に登録され、<br />
レコードが有るものは更新されます。 <br />
&nbsp;<br />
<asp:Label ID="LabelError" runat="server" Text="LabelError"></asp:Label><br />
===== </div>
</form>
</body>
</html>
```

## register.aspx.designer.vb

```
'-----  
' <auto-generated>  
' このコードはツールによって生成されました。  
' ランタイム バージョン:2.0.50727.1433  
'  
' このファイルへの変更は、以下の状況下で不正な動作の原因になったり、  
' コードが再生成されるときに損失したりします。  
' </auto-generated>  
'-----  
  
Option Strict Off  
Option Explicit On  
  
'''<summary>  
'''register クラス。  
'''</summary>  
'''<remarks>  
''' 自動生成されたクラス。  
'''</remarks>  
Partial Public Class register  
  
    '''<summary>  
    '''form1 コントロール。  
    '''</summary>  
    '''<remarks>  
    ''' 自動生成されたフィールド。  
    ''' 変更するには、フィールドの宣言をデザイナ ファイルから分離コード ファイルに移動します。  
    '''</remarks>  
    Protected WithEvents form1 As Global.System.Web.UI.HtmlControls.HtmlForm  
  
    '''<summary>  
    '''TextBox1 コントロール。  
    '''</summary>  
    '''<remarks>  
    ''' 自動生成されたフィールド。  
    ''' 変更するには、フィールドの宣言をデザイナ ファイルから分離コード ファイルに移動します。  
    '''</remarks>  
    Protected WithEvents TextBox1 As Global.System.Web.UI.WebControls.TextBox  
  
    '''<summary>  
    '''TextBox2 コントロール。  
    '''</summary>  
    '''<remarks>  
    ''' 自動生成されたフィールド。  
    ''' 変更するには、フィールドの宣言をデザイナ ファイルから分離コード ファイルに移動します。  
    '''</remarks>  
    Protected WithEvents TextBox2 As Global.System.Web.UI.WebControls.TextBox  
  
    '''<summary>  
    '''DropDownListDepartment コントロール。  
    '''</summary>  
    '''<remarks>  
    ''' 自動生成されたフィールド。  
    ''' 変更するには、フィールドの宣言をデザイナ ファイルから分離コード ファイルに移動します。  
    '''</remarks>  
    Protected WithEvents DropDownListDepartment As Global.System.Web.UI.WebControls.DropDownList  
  
    '''<summary>  
    '''DropDownListGroupes コントロール。  
    '''</summary>  
    '''<remarks>  
    ''' 自動生成されたフィールド。  
    ''' 変更するには、フィールドの宣言をデザイナ ファイルから分離コード ファイルに移動します。  
    '''</remarks>  
    Protected WithEvents DropDownListGroups As Global.System.Web.UI.WebControls.DropDownList  
  
    '''<summary>  
    '''DropDownListValid コントロール。  
    '''</summary>  
    '''<remarks>  
    ''' 自動生成されたフィールド。  
    ''' 変更するには、フィールドの宣言をデザイナ ファイルから分離コード ファイルに移動します。  
    '''</remarks>  
    Protected WithEvents DropDownListValid As Global.System.Web.UI.WebControls.DropDownList  
  
    '''<summary>  
    '''ButtonCheck コントロール。  
    '''</summary>  
    '''<remarks>  
    ''' 自動生成されたフィールド。  
    ''' 変更するには、フィールドの宣言をデザイナ ファイルから分離コード ファイルに移動します。  
    '''</remarks>  
    Protected WithEvents ButtonCheck As Global.System.Web.UI.WebControls.Button  
  
    '''<summary>  
    '''Label1 コントロール。  
    '''</summary>  
    '''<remarks>  
    ''' 自動生成されたフィールド。  
    ''' 変更するには、フィールドの宣言をデザイナ ファイルから分離コード ファイルに移動します。  
    '''</remarks>  
    Protected WithEvents Label1 As Global.System.Web.UI.WebControls.Label  
  
    '''<summary>  
    '''ButtonRegister コントロール。  
    '''</summary>  
    '''<remarks>  
    ''' 自動生成されたフィールド。  
    ''' 変更するには、フィールドの宣言をデザイナ ファイルから分離コード ファイルに移動します。  
    '''</remarks>  
    Protected WithEvents ButtonRegister As Global.System.Web.UI.WebControls.Button  
  
    '''<summary>  
    '''LabelError コントロール。  
    '''</summary>  
    '''<remarks>  
    ''' 自動生成されたフィールド。  
    ''' 変更するには、フィールドの宣言をデザイナ ファイルから分離コード ファイルに移動します。  
    '''</remarks>  
    Protected WithEvents LabelError As Global.System.Web.UI.WebControls.Label  
End Class
```

## web.config

```
<?xml version="1.0"?>
<!--
  メモ: このファイルを手動で編集する代わりに、Web 管理ツールを使用してアプリケーションの設定を構成することができます。Visual Studio の [Web サイト] メニューにある [ASP.NET 構成] オプションから設定を行ってください。設定およびコマンドの一覧は、通常 \Windows\Microsoft.Net\Framework\v2.0.50727\Config にある machine.config.comments で確認できます。
-->
<configuration>
  <appSettings/>

  <system.web>
    <!--
      デバッグ シンボルをコンパイルされたページに挿入するには、compilation debug="true" に設定します。この設定はパフォーマンスに影響するため、開発時のみこの値を true に設定してください。

      Visual Basic オプション:
      データの損失が発生する可能性のあるすべてのデータ型の変換を無効にするには、strict="true" に設定します。
      すべての変数の定義を強制するためには、explicit="true" に設定します。
    -->
    <compilation debug="true" strict="false" explicit="true">
      <assemblies>
        <add assembly="System.DirectoryServices, Version=2.0.0.0, Culture=neutral, PublicKeyToken=B03F5F7F11D50A3A"/>
      </assemblies>
    </compilation>

    <pages>
    <namespaces>
    <clear/>
    <add namespace="System"/>
    <add namespace="System.Collections"/>
    <add namespace="System.Collections.Specialized"/>
    <add namespace="System.Configuration"/>
    <add namespace="System.Text"/>
    <add namespace="System.Text.RegularExpressions"/>
    <add namespace="System.Web"/>
    <add namespace="System.Web.Caching"/>
    <add namespace="System.Web.SessionState"/>
    <add namespace="System.Web.Security"/>
    <add namespace="System.Web.Profile"/>
    <add namespace="System.Web.UI"/>
    <add namespace="System.Web.UI.WebControls"/>
    <add namespace="System.Web.UI.WebControls.WebParts"/>
    <add namespace="System.Web.UI.HtmlControls"/>
    </namespaces>
    </pages>
    <!--
      <authentication> セクションは、ユーザーを識別するために、ASP.NET で使用されるセキュリティ認証モードの構成を有効にします。
    -->
    <authentication mode="Windows"/>
    <!--
      <customErrors> セクションは、要求の実行中にハンドルされていないエラーが発生した場合の処理方法の構成を有効にします。具体的には、開発者が HTML エラー ページをスタック トレースのエラーの代わりに表示するように構成することを可能にします。
    -->
    <customErrors mode="On" defaultRedirect="GenericErrorPage.htm">
      <error statusCode="403" redirect="NoAccess.htm" />
      <error statusCode="404" redirect="FileNotFound.htm" />
    </customErrors>
    <!--
      偽装の設定。これをやらないと ID が SQL Server に伝達されず、コンピュータアカウントで認証されてしまう
    -->
    <identity impersonate="true"/>
  </system.web>
</configuration>
```