CCMS/CTR Security

CCMS/CTR security is based on Windows Domain-User (Windows Identity) authentication system, version control system, and custom authorization system.

Classes used to authenticate a Windows domain-user:

WindowsIdentity (Inherits System.Object, Member of System.Security.Principal)

DirectoryEntry (Inherits System.ComponentModel.Component, Member of System.DirectoryServices)

Codes used to authenticate a windows user:

Dim flagLoginSuccess As Boolean = False

Dim ident As WindowsIdentity = WindowsIdentity.GetCurrent()

Dim entry As New DirectoryEntry

entry.Username = ident.Name

entry.Password = txtPass.Text.Trim()

Try

If CommonAppSet.IsByPass = False Then

Dim obj As Object = entry.NativeObject

End If

flagLoginSuccess = True

Catch ex As Exception

End Try

Classe used for version control system:

Assembly (Inherits System.Object, Member of System.Reflection)

Stored procedure used for version control:

CMS\_Version\_GetDetail

Codes used to control version:

Try

Dim db As New SqlDatabase(CommonAppSet.ConnStr)

Dim commProc As DbCommand = db.GetStoredProcCommand("CMS\_Version\_GetDetail")

Dim dt As New DataTable

dt = db.ExecuteDataSet(commProc).Tables(0)

If dt.Rows.Count > 0 Then

If (dt.Rows(0)("CHK").ToString() = "1") Then

Dim dbMinMajor As Integer = dt.Rows(0)("MIN\_MAJOR")

Dim dbMinMinor As Integer = dt.Rows(0)("MIN\_MINOR")

Dim dbMinBuild As Integer = dt.Rows(0)("MIN\_BUILD")

Dim dbMaxMajor As Integer = dt.Rows(0)("MAX\_MAJOR")

Dim dbMaxMinor As Integer = dt.Rows(0)("MAX\_MINOR")

Dim dbMaxBuild As Integer = dt.Rows(0)("MAX\_BUILD")

Dim verApp As Version = Assembly.GetExecutingAssembly().GetName().Version

Dim verMin As New Version(dbMinMajor, dbMinMinor, dbMinBuild, 0)

Dim verMax As New Version(dbMaxMajor, dbMaxMinor, dbMaxBuild, 0)

If (verMin <= verApp And verMax >= verApp) Then

Return True

End If

MessageBox.Show("Application version incompatible." + Environment.NewLine + \_

"Your app ver: " + Application.ProductVersion + Environment.NewLine + \_

"Please contact with your administrator to continue.", "Version Incompatible!!", MessageBoxButtons.OK, MessageBoxIcon.Error)

Else

Return True

End If

Else

MessageBox.Show("No version information found", "Version Error!!", MessageBoxButtons.OK, MessageBoxIcon.Error)

End If

Catch ex As SqlClient.SqlException

If ex.ErrorCode = -2146232060 Then

MessageBox.Show("You are not authorized!!", "Version Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

Else

MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

End If

Exit Function

Catch ex As Exception

MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

Exit Function

End Try

Return False

Stored procedures used for custom authorization:

CMS\_Users\_GetDetailByCode

CMS\_Users\_TryLogin

CMS\_Users\_InvalidLogin

CMS\_Users\_FunFormPermission

CMS\_Users\_FunMenuPermission

Codes used for custom authorization:

If flagLoginSuccess = True Then

Try

Dim db As New SqlDatabase(CommonAppSet.ConnStr)

Dim commProc As DbCommand = db.GetStoredProcCommand("CMS\_Users\_GetDetailByCode")

commProc.Parameters.Clear()

db.AddInParameter(commProc, "@USERS\_ID", DbType.String, CommonAppSet.User)

Dim dt As New DataTable

dt = db.ExecuteDataSet(commProc).Tables(0)

If dt.Rows.Count > 0 Then

If (dt.Rows(0)("USER\_STAT").ToString() = "E") Then

commProc = db.GetStoredProcCommand("CMS\_Users\_TryLogin")

commProc.Parameters.Clear()

db.AddParameter(commProc, "@PROC\_RET\_VAL", DbType.Int32, ParameterDirection.ReturnValue, DBNull.Value.ToString(), DataRowVersion.Default, DBNull.Value)

Dim result As Integer

db.ExecuteNonQuery(commProc)

result = db.GetParameterValue(commProc, "@PROC\_RET\_VAL")

If result = 0 Then

If NullHelper.ToBool(dt.Rows(0)("IS\_ADMIN")) = True Then

CommonAppSet.IsAdmin = True

Else

CommonAppSet.IsAdmin = False

End If

Logger.system\_log("Successful login: " + CommonAppSet.User)

Return True

ElseIf result = 1 Then

MessageBox.Show("System Error. Login Denied!!", "Systen Error!!", MessageBoxButtons.OK, MessageBoxIcon.Error)

Logger.system\_log("User: " + CommonAppSet.User + " Login Failed. System Error")

ElseIf result = 4 Then

MessageBox.Show("! Account not exist !" + Environment.NewLine + \_

"Contact with your administrator", "Login Failed", MessageBoxButtons.OK, MessageBoxIcon.Error)

Logger.system\_log("User: " + CommonAppSet.User + " not exists. Login Failed.")

ElseIf result = 8 Then

MessageBox.Show("You acccunt has been disabled. Login Denied!!", "Systen Error!!", MessageBoxButtons.OK, MessageBoxIcon.Error)

Logger.system\_log("User: " + CommonAppSet.User + " Login Failed. Account is locked.")

Else

MessageBox.Show("System Error. Login Denied!!", "Systen Error!!", MessageBoxButtons.OK, MessageBoxIcon.Error)

Logger.system\_log("User: " + CommonAppSet.User + " Login Failed. System Error")

End If

ElseIf (dt.Rows(0)("USER\_STAT").ToString() = "D") Then

MessageBox.Show("Your ID is locked!!", "Login Error!!", MessageBoxButtons.OK, MessageBoxIcon.Error)

Logger.system\_log("User: " + CommonAppSet.User + " Login Failed. Your ID is locked.")

Else

MessageBox.Show("Your ID is corrupted!!", "Login Error!!", MessageBoxButtons.OK, MessageBoxIcon.Error)

Logger.system\_log("User: " + CommonAppSet.User + " Login Failed. ID record corrupted")

End If

Else

MessageBox.Show("You are not authorized!!", "Login Error!!", MessageBoxButtons.OK, MessageBoxIcon.Error)

Logger.system\_log("User: " + CommonAppSet.User + " Login Failed. ID is not maintained.")

End If

Catch ex As Exception

MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

End Try

Else

Try

Dim db As New SqlDatabase(CommonAppSet.ConnStr)

Dim commProc As DbCommand = db.GetStoredProcCommand("CMS\_Users\_InvalidLogin")

commProc.Parameters.Clear()

db.AddParameter(commProc, "@PROC\_RET\_VAL", DbType.Int32, ParameterDirection.ReturnValue, DBNull.Value.ToString(), DataRowVersion.Default, DBNull.Value)

Dim result As Integer

db.ExecuteNonQuery(commProc)

result = db.GetParameterValue(commProc, "@PROC\_RET\_VAL")

If result = 0 Then

MessageBox.Show("Invalid User/Password", "Login Failed", MessageBoxButtons.OK, MessageBoxIcon.Error)

Logger.system\_log("User: " + CommonAppSet.User + " Login Failed. Invalid user/password.")

txtPass.SelectAll()

txtPass.Focus()

ElseIf result = 4 Then

MessageBox.Show("! Account not exist !" + Environment.NewLine + \_

"Contact with your administrator", "Login Failed", MessageBoxButtons.OK, MessageBoxIcon.Error)

Logger.system\_log("User: " + CommonAppSet.User + " not exists. Login Failed.")

txtPass.SelectAll()

txtPass.Focus()

ElseIf result = 8 Then

MessageBox.Show("! Account is locked !" + Environment.NewLine + \_

"Contact with your administrator", "Login Failed", MessageBoxButtons.OK, MessageBoxIcon.Error)

txtPass.SelectAll()

txtPass.Focus()

Else

MessageBox.Show("Unspecified error occured" + Environment.NewLine + \_

"Contact with your administrator", "Invalid Login", MessageBoxButtons.OK, MessageBoxIcon.Error)

Logger.system\_log("User: " + CommonAppSet.User + " Login Failed. Unspecififed Error Occured.")

txtPass.SelectAll()

txtPass.Focus()

End If

Catch ex As Exception

MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

End Try

End If

Return False

Codes used for Form Permission:

Try

Dim user = CommonAppSet.User

Dim db As New SqlDatabase(CommonAppSet.ConnStr)

Dim ds As New DataSet

Dim comProc As DbCommand = db.GetStoredProcCommand("CMS\_Users\_FunFormPermission")

db.AddInParameter(comProc, "@USERS\_ID", DbType.String, user)

db.AddInParameter(comProc, "@FORMS\_NAME", DbType.String, FormName)

'db.AddInParameter(comProc, "@USERS\_ID", DbType.String, "fahad")

'db.AddInParameter(comProc, "@FORMS\_NAME", DbType.String, "rere")

CommonAppSet.DebugMessage = "Perm Proc, Formname val: " + FormName + \_

", User: " + user

ds = db.ExecuteDataSet(comProc)

If ds.Tables(0).Rows.Count > 0 Then

'FormName = ds.Tables(0).Rows(0)("FORMS\_NAME").ToString()

CommonAppSet.DebugMessage = CommonAppSet.DebugMessage + \_

Environment.NewLine + ", show: " + NullHelper.ObjectToString(ds.Tables(0).Rows(0)("IS\_SHOW"))

\_IsShow = ds.Tables(0).Rows(0)("IS\_SHOW").ToString()

CommonAppSet.DebugMessage = CommonAppSet.DebugMessage + \_

Environment.NewLine + ", new: " + NullHelper.ObjectToString(ds.Tables(0).Rows(0)("IS\_NEW"))

\_IsNew = ds.Tables(0).Rows(0)("IS\_NEW").ToString()

CommonAppSet.DebugMessage = CommonAppSet.DebugMessage + \_

Environment.NewLine + ", unlock: " + NullHelper.ObjectToString(ds.Tables(0).Rows(0)("IS\_UNLOCK"))

\_IsUnlock = ds.Tables(0).Rows(0)("IS\_UNLOCK").ToString()

CommonAppSet.DebugMessage = CommonAppSet.DebugMessage + \_

Environment.NewLine + ", auth: " + NullHelper.ObjectToString(ds.Tables(0).Rows(0)("IS\_AUTHORIZER"))

\_IsAuth = ds.Tables(0).Rows(0)("IS\_AUTHORIZER").ToString()

CommonAppSet.DebugMessage = CommonAppSet.DebugMessage + \_

Environment.NewLine + ", delete: " + NullHelper.ObjectToString(ds.Tables(0).Rows(0)("IS\_DELETE"))

\_IsDelete = ds.Tables(0).Rows(0)("IS\_DELETE").ToString()

CommonAppSet.DebugMessage = CommonAppSet.DebugMessage + \_

Environment.NewLine + ", save: " + NullHelper.ObjectToString(ds.Tables(0).Rows(0)("IS\_SAVE"))

\_IsSave = ds.Tables(0).Rows(0)("IS\_SAVE").ToString()

End If

Catch ex As Exception

MessageBox.Show("Debug Level: " + CommonAppSet.DebugLevel + \_

Environment.NewLine + "Debug Message: " + CommonAppSet.DebugMessage + \_

Environment.NewLine + "App Message: " + ex.Message, "Security Class Error")

End Try

-----

If IsAdmin = True Then

\_IsShow = True

\_IsNew = True

\_IsUnlock = True

\_IsAuth = True

\_IsDelete = True

\_IsSave = True

Else

Dim user = CommonAppSet.User

Dim db As New SqlDatabase(CommonAppSet.ConnStr)

Dim ds As New DataSet

Dim comProc As DbCommand = db.GetStoredProcCommand("CMS\_Users\_FunFormPermission")

db.AddInParameter(comProc, "@USERS\_ID", DbType.String, user)

db.AddInParameter(comProc, "@FORMS\_NAME", DbType.String, FormName)

ds = db.ExecuteDataSet(comProc)

If ds.Tables(0).Rows.Count > 0 Then

'FormName = ds.Tables(0).Rows(0)("FORMS\_NAME").ToString()

\_IsShow = ds.Tables(0).Rows(0)("IS\_SHOW").ToString()

\_IsNew = ds.Tables(0).Rows(0)("IS\_NEW").ToString()

\_IsUnlock = ds.Tables(0).Rows(0)("IS\_UNLOCK").ToString()

\_IsAuth = ds.Tables(0).Rows(0)("IS\_AUTHORIZER").ToString()

\_IsDelete = ds.Tables(0).Rows(0)("IS\_DELETE").ToString()

\_IsSave = ds.Tables(0).Rows(0)("IS\_SAVE").ToString()

End If

End If

If CommonAppSet.IsByPass = True Then

\_IsShow = True

\_IsNew = True

\_IsUnlock = True

\_IsAuth = True

\_IsDelete = True

\_IsSave = True

End If

Codes used for Menu Permission:

Dim comProc As DbCommand = db.GetStoredProcCommand("CMS\_Users\_FunMenuPermission")

db.AddInParameter(comProc, "@USERS\_ID", DbType.String, user)

ds = db.ExecuteDataSet(comProc)

For i = 0 To ds.Tables(0).Rows.Count - 1

\_menuname = ds.Tables(0).Rows(i)("MENU\_NAME").ToString()

\_isvisible = ds.Tables(0).Rows(i)("IS\_VISIBLE").ToString()

\_enable = ds.Tables(0).Rows(i)("IS\_ENABLE").ToString

MenuPerm()

Next