****

**DATAVISION IMAGE LLC**

**906 Via De Angeles**

**San Clemente, CA 92672**

**info@datavisionimage.com**

**Technical Instructions for using the DATAVISION Corporate Encryption API**

**Contents of the DLL**

The Corporate Encryption DLL has three commands:

**GetPassword**(ByVal MyPasswordIn As String) As String

**EncryptString**(ByVal MyString As String) As String

**DeEncryptString**(ByVal MyString As String) As String

**Discussion**

Corporate Encryption is a Windows API that can easily be added to any Windows program including Visual Studio programs such as VB.Net, C#.Net, ASP.NET to allow the program to internally encrypt and decrypt data with a single, simple command.

Sensitive data such as connection strings, passwords, sensitive parameters, lists, etc. can be internally encrypted and therefore will be unavailable to anyone trying to hack or use parts of the program to gain access to lists, connections string, and so forth.

The Windows API has three commands that can be interactively used by the programmer to protect exposed data.

The Windows API can also be used to construct encryption programs that can only be deciphered by a recipient of the Windows API along with the password used for encryption. This practice is referred to as salted password protection. While widely used in angular JS type encryption, the use in Windows programs is more complex. This Windows API greatly simplifies the process.

The kit contains the following items:

**Encrypt Two DLL**

**Install program for test demo program**

**Source code for test demo program**

**Programmer instructions for deploying and using the DLL**

Here are the results of the test program

At Start:



Failure to enter a password will result in an error:



When a string is entered the ADD button encrypts it so the screen looks like this:



The code to encrypt is very simple. Notice how the password is passed at the start of the program and when the Decrypt command is called.

MyPassword = Trim(txtPassword.Text)

 If Trim(txtPassword.Text) = "" Then

 MsgBox("You Must Enter A Password For Encryption")

 txtPassword.Focus()

 Exit Sub

 End If

 MyPassword = MyEncryption.GetPassword(MyPassword)

 TextToAdd = txtEnterText.Text

 EncryptedText = MyEncryption.EncryptString(TextToAdd, MyPassword)

Notice the warning message is generated by the application. The DLL will return the phrase “PASSWORD REQUIRED” instead of the encrypted text if the password is missing.

The application allows you to add items to the list



Press the Decrypt button to get the clear text. Here is the code used by the Decrypt button

MyPassword = Trim(txtPassword.Text)

 If Trim(txtPassword.Text) = "" Then

 MsgBox("You Must Enter A Password For Encryption")

 txtPassword.Focus()

 Exit Sub

 End If

 MyPassword = MyEncryption.GetPassword(MyPassword)

 For I = 0 To (MyCounter) - 1

 TextToDecrypt = lstNames.Items(I)

 MyDecryptedText = MyEncryption.DeEncryptString(TextToDecrypt, MyPassword)

 lstNames.Items.Add(MyDecryptedText)

 Next

Notice the warning is given by the application program. The DLL will return this screen if the password is missing or incorrect:



This will result in the following:



This message is produced by the DLL and will repeat for each line with an incorrect password

If the correct password is used the encrypted text is replaced by clear text



Using these simple commands, the programmer can easily protect connection strings, lists, passwords and so forth with only a line or two of code.

**Source Code:**

Public Class frmMain

 Public MyEncryption As New EncryptionTwo.EncryptTextTwo

 'this is how to instantiate the DLL

 Dim MyPassword As String

 Private Sub cmdEnd\_Click(sender As Object, e As EventArgs) Handles cmdEnd.Click

 End

 End Sub

 Private Sub frmMain\_Load(sender As Object, e As EventArgs) Handles Me.Load

 lstNames.Items.Clear()

 End Sub

 Private Sub cmdAdd\_Click(sender As Object, e As EventArgs) Handles cmdAdd.Click

 'this sub adds strings to the list box and encrypts them as you go

 Dim TextToAdd As String = ""

 Dim EncryptedText As String = ""

 Dim NumberOfItems As Integer = 0

 MyPassword = Trim(txtPassword.Text)

 If Trim(txtPassword.Text) = "" Then

 MsgBox("You Must Enter A Password For Encryption")

 txtPassword.Focus()

 Exit Sub

 End If

 MyPassword = MyEncryption.GetPassword(MyPassword)

 TextToAdd = txtEnterText.Text

 EncryptedText = MyEncryption.EncryptString(TextToAdd, MyPassword)

 lstNames.Items.Add(EncryptedText)

 txtEnterText.Text = ""

 NumberOfItems = lstNames.Items.Count

 lblCount.Text = NumberOfItems.ToString

 txtEnterText.Focus()

 End Sub

 Private Sub cmdClear\_Click(sender As Object, e As EventArgs) Handles cmdClear.Click

 'clears the data

 txtEnterText.Text = ""

 lstNames.Items.Clear()

 lblCount.Text = (lstNames.Items.Count).ToString

 End Sub

 Private Sub txtPassword\_LostFocus(sender As Object, e As EventArgs) Handles txtPassword.LostFocus

 'this sub check that a password has been entered

 If Trim(txtPassword.Text) = "" Then

 MsgBox("You Must Enter A Password For Encryption")

 txtPassword.Focus()

 Exit Sub

 End If

 End Sub

 Private Sub cmdDeCrypt\_Click(sender As Object, e As EventArgs) Handles cmdDeCrypt.Click

 'this sub decrypts the strings in the list box and replaces the encrypted text with clear text

 Dim MyDecryptedText As String = ""

 Dim TextToDecrypt As String = ""

 'lstNames.Items.Add("")

 'lstNames.Items.Add("TRANSLATED VALUES")

 Dim MyCounter As Integer = lstNames.Items.Count

 Dim ClearCounter As Integer = 0

 For I = 0 To (MyCounter) - 1

 TextToDecrypt = lstNames.Items(I)

 MyDecryptedText = MyEncryption.DeEncryptString(TextToDecrypt, MyPassword)

 lstNames.Items.Add(MyDecryptedText)

 ClearCounter = ClearCounter + 1

 Next

 For i = ClearCounter - 1 To 0 Step -1

 lstNames.Items.RemoveAt(i)

 Next

 lstNames.Refresh()

 End Sub

End Class