Success on the Web: String Randomization And Encryption In Classic ASP

**TUESDAY, JUNE 24, 2008**

|  |  |  |
| --- | --- | --- |
|  | **String Randomization And Encryption In Classic ASP** |  |
| Every now and then the need comes up to create a random string of characters. Most frequently this need is brought on while implementing a "password reset" type of feature. I don't like sending real passwords back to users, even if they are stored in plain text (which hopefully they are not, but I have continually been surprised at the high profile clients that will do this) as it implies to the visitor we can actually see their password, which is, in my mind, a bad message to send. I prefer to reset it and sending them the new, randomly generated password and letting them reset it manually to whatever they want once logged in. For the record, I always encourage clients to encrypt passwords, even if it's a simple home-brewed encryption method, any is better than none, and I always practice that preaching myself. I almost always run password through some rediculous series of one way hashes (not really one way as most have been broken), string manipulation and other processes to obfuscate the data as much as possible. So, to get back on topic, here is a quick little function to create a random string of characters of any length you pass to the function. Function createRandomString(pwLength) Dim charArray,arrayLength,pw,i charArray = "ABCDEFGHIJKLMNPQRSTUVWXYZ0123456789" arrayLength = Len(charArray) pw = "" Randomize For i = 1 To pwLength pw = pw & Mid(charArray, 1 + Int(Rnd \* arrayLength), 1) Next createRandomString = pw End Function Remember, while this is a random string, before storing it, be sure to run it through the same encryption functions that you would any user password if in fact it is being used as a password. Classic ASP has a lack of built in encryption functions, or one way hashes like PHP does, but, it is quite simple to create a small ASCII replacement function that will simply replace each character with a character a set number of ASCII values down the chain (in this case 8 values down), and that also makes a simple decrypt function available by reversing the process. Function encodeString(input) Dim newStr : newStr = "" for i = 1 to len(input) newStr = newStr & chr((asc(mid(input,i,1))+8)) next encodeString = newStr End Function Function decodeString(input) Dim oldStr : oldStr = "" for i = 1 to len(input) oldStr = oldStr & chr((asc(mid(input,i,1))-8)) next decodeString = oldStr End Function Obviously, I would recommend finding a better, more robust, way to encrypt passwords, but, at the end of the day, and like I said before, anything is better than nothing, and these can work in a pinch. |
| Posted by dB Masters Labels: Active Server Pages, Web Site Security |
|  |  |  |

http://successontheweb.blogspot.com/2008/06/string-randomization-and-e...

1 of 2 12/24/2009 2:14 PM

Success on the Web: String Randomization And Encryption In Classic ASP

**2 comments:**

http://successontheweb.blogspot.com/2008/06/string-randomization-and-e...

**zain said...**

Its very good, But the length of encoded string is the same, I mean if i pass only 7 in Arguments to Encode, so i always get the '?' character, how about if the returned encoded string is of minimum 8 characters.. please see for this...

July 19, 2008 12:20 PM

**Anonymous said...**

Author: Simon Carlisle

Private Function EncryptText(ByVal strEncryptionKey, ByVal strTextToEncrypt) If encrypt = 1 Then ' Declare variables Dim outer, inner, Key, strTemp ' For each character in strEncryptionKey For outer = 1 To Len(strEncryptionKey) ' Get a character to use as our encryption ' key in this iteration of the OUTER loop key = Asc(Mid(strEncryptionKey, outer, 1)) ' For each character in strTextToEncrypt For inner = 1 To Len(strTextToEncrypt) ' Update our encrypted text strTemp = strTemp & Chr(Asc(Mid(strTextToEncrypt, inner, 1)) Xor key) ' Change our encryption key to mix things up in the INNER loop. key = (key + Len(strEncryptionKey)) Mod 256 Next ' Update the strTextToEncrypt variable before ' the next iteration of the OUTER loop strTextToEncrypt = strTemp ' Reset strTemp for the next iteration of the OUTER loop. strTemp = "" Next End If ' Assign the value of the encrypted text to the function name ' so we can return the value to the caller EncryptText = strTextToEncrypt End Function

12/24/2009 2:14 PM2 of 2