

# Forum:[Hack Forums](#)

Thread: [ActualRand\(\)](#)

**PapaBeans** - September 1, 2014, 12:14 am

Hello fellow coders of HF!

In my couple years of coding I've come to realize that the default "Random", doesn't return a random. It tends to return very close values if executed within short time periods. I decided to take action, and make a truly 100% random Random function. Me and my friend (mostly him) came up with this algorithm.

Pictures are provided for everyone

If you are L33t or Ub3r then PM me for straight source ( name the PM Random)

Please leave feedback, it is much appreciated. ?

EDIT:: So you guys said it was stolen, I didn't know. As I said he did a lot of the work, I didn't know that he took it from a site. I apologize.

**CottleSilver** - September 1, 2014, 1:08 am

You didn't do this:

<http://www.vbforums.com/showthread.php?357926-quot-True-Random-quot-Number-Generator>.

The code from the link above (Attached Files):

Code Public Class TrueRandom

.....

"Use this class at will ""

"but please leave this note attached""

" - Phill64 (Stack Overflow) ""

.....

"True Random

.....

"This random class uses 4 factors most likely to change:

"System Timer

"System Memory (RAM)

"CPU Usage

"Mouse Position

.....

"Why is this class true random and the pre-packaged one isn't you ask?

"Because classic uses just the timer as a seed, and a pattern to generate from then on

"Whereas, this class only follows patterns for a limited time(long enough for new data to come in)

"and the fact it uses RAM and the Mouse Position means it is effected by unknown sources.

"the system clock spins circles, so it will of course create a pattern

"a user moving his mouse is COMPLETELY unpredictable

"RAM is effected continuously in ways that can never be predicted(within reason)

"CPU performance is effected by everything running and even the weather! thus the CPU can also never be accurately predicted

.....

"How did I test the results?

.....

"by creating 2 classes of the classic Random

"and 2 classes of TrueRandom

"then making all 4 create a list of random numbers

"classic random generated 2 IDENTICAL lists every time i tested

"whereas the 2 TrueRandom classes would create 2 different lists

.....

"Conclusion:

"this class, created under all the same conditions returns different results

"that.. is truly random.

'these variables make a temporary pattern,

'long enough for data to change

Private betTick As Double 'between ticks

Private lastRAM As Decimal 'last value from memory

Private tempPat As Integer 'small pattern iterater

Public Function GetNext(ByRef lower As Integer, ByRef upper As Integer) As Integer

    'get system timer

    betTick = (betTick + Now.Ticks) / 2 'between ticks, gradually move towards next tick of the system clock

    'get system RAM

    Dim RAM As Decimal 'where we'll put information taken from RAM

    Dim i As Integer

    For i = 0 To 40

        RAM = ((RAM + Threading.Thread.VolatileRead(betTick + i)) / 2) 'retrieve next byte and average with last

    Next

    RAM = Mid((((lastRAM + RAM) / 2).ToString, 5) 'average RAM just retrieved with last RAM retrieved

    lastRAM = RAM

    'apply timer,RAM,CPU and Mouse

    tempPat += 1

    If tempPat > 20 Then tempPat = 1

    Dim v As Double = (RAM / tempPat) + ((betTick / 100) + CPUUsage() + (Cursor.Position.X / Cursor.Position.Y))

    Dim rand As Decimal = (v / 1000) - Math.Floor(v / 1000)

    'ratio the random decimal number to the required bounds

    Return CLng(lower + (rand \* (upper - lower)))

End Function

"CPU Usage code from: [http://www.theserverside.net/discussions/thread.tss?thread\\_id=26337](http://www.theserverside.net/discussions/thread.tss?thread_id=26337)

Private Function CPUUsage() As Double

    Dim myCounter As System.Diagnostics.PerformanceCounter = New

System.Diagnostics.PerformanceCounter

    myCounter.CategoryName = "Processor"

    myCounter.CounterName = "% Processor Time"

    myCounter.InstanceName = "\_Total"

    Return myCounter.RawValue

End Function

End ClassI ask everyone if he didn't steal anything else... ?

PapaBeans - September 1, 2014, 1:33 am

(Sep 1, 2014 01:08 AM)CottleSilver Wrote: [\[CITE\]](#) You didn't do this:

<http://www.vbforums.com/showthread.php?357926-quot-True-Random-quot-Number-Generator>.

The code from the link above (Attached Files):

Code Public Class TrueRandom

.....

"Use this class at will        ""

"but please leave this note attached""

" - Phill64 (Stack Overflow)        ""

.....

"True Random

.....

"This random class uses 4 factors most likely to change:

"System Timer

"System Memory (RAM)

"CPU Usage

"Mouse Position

.....

"Why is this class true random and the pre-packaged one isn't you ask?

"Because classic uses just the timer as a seed, and a pattern to generate from then on

"Whereas, this class only follows patterns for a limited time(long enough for new data to come in)

"and the fact it uses RAM and the Mouse Position means it is effected by unknown sources.

"the system clock spins circles, so it will of course create a pattern

"a user moving his mouse is COMPLETELY unpredictable

"RAM is effected continuously in ways that can never be predicted(within reason)

"CPU performance is effected by everything running and even the weather! thus the CPU can also never be accurately predicted

.....

"How did I test the results?

.....

"by creating 2 classes of the classic Random

"and 2 classes of TrueRandom

"then making all 4 create a list of random numbers

"classic random generated 2 IDENTICAL lists every time i tested

"whereas the 2 TrueRandom classes would create 2 different lists

.....

"Conclusion:

"this class, created under all the same conditions returns different results

"that.. is truely random.

'these variables make a temporary pattern,

'long enough for data to change

Private betTick As Double 'between ticks

Private lastRAM As Decimal 'last value from memory

Private tempPat As Integer 'small pattern iterater

Public Function GetNext(ByRef lower As Integer, ByRef upper As Integer) As Integer

  'get system timer

  betTick = (betTick + Now.Ticks) / 2 'between ticks, gradually move towards next tick of the system clock

```

'get system RAM
Dim RAM As Decimal 'where we'll put information taken from RAM
Dim i As Integer
For i = 0 To 40
    RAM = ((RAM + Threading.Thread.VolatileRead(betTick + i)) / 2) 'retrieve next byte and
average with last
Next
RAM = Mid((((lastRAM + RAM) / 2).ToString, 5) 'average RAM just retrieved with last RAM
retrieved
lastRAM = RAM

'apply timer, RAM, CPU and Mouse
tempPat += 1
If tempPat > 20 Then tempPat = 1
Dim v As Double = (RAM / tempPat) + ((betTick / 100) + CPUUsage() + (Cursor.Position.X /
Cursor.Position.Y))
Dim rand As Decimal = (v / 1000) - Math.Floor(v / 1000)

'ratio the random decimal number to the required bounds
Return CLng(lower + (rand * (upper - lower)))
End Function

"CPU Usage code from: http://www.theserverside.net/discussions/thread.tss?thread\_id=26337
Private Function CPUUsage() As Double
    Dim myCounter As System.Diagnostics.PerformanceCounter = New
System.Diagnostics.PerformanceCounter
    myCounter.CategoryName = "Processor"
    myCounter.CounterName = "% Processor Time"
    myCounter.InstanceName = "_Total"
    Return myCounter.RawValue
End Function

```

End Class I ask everyone if he didn't steal anything else... ?

I apologize, I shouldve checked if he took it from an online source. Anyway, enjoy.

**He B TeMy** - September 1, 2014, 3:35 am

Lol, when you're caught stealing, you edit the post saying it was you and your friend, how clever.

**CottleSilver** - September 1, 2014, 3:49 am

(Sep 1, 2014 03:35 AM)HeBTeMy Wrote: [\[CITE\]](#) Lol, when you're caught stealing, you edit the post saying it was you and your friend, how clever.

Yea coz somebody waits 7 years before posting a code... The op stole a code and posted as it's own, the end.?

**zeroexp** - September 1, 2014, 6:54 am

Random is random enough already, trust me. I've spent hours examining the randomness of that and there is absolutely no need for an inprovement, after all, nothing is absolutely random. No need for this.

**ubbelol** - September 1, 2014, 7:23 am

This is far from "100% random". The built-in Random class works perfectly fine if you use it correctly, which you obviously don't. Additionally your method is really slow compared to the .NET implementation.

**LordCoder** - September 1, 2014, 7:25 am

That's not 100% random. This is pseudo random, the only random way a computer can do. And the Random class is based on the time the computer was on (Environment.TickCount).

```
Code Public Sub New()  
    Me.New(Environment.TickCount)  
End Sub
```

You can try to remove that similar with different seed. Just use any hardware property as value and that's all. No need to create a new function when it's already done.

<http://en.wikipedia.org/wiki/Pseudorandomness>

<http://stackoverflow.com/questions/12076358/confusion-about-series-of-pure-random-numbers>

**ubbelol** - September 1, 2014, 7:36 am

Did some testing just for fun, and to give you a representation of why trying to implement stuff like this yourself is not a good idea I'll show you a quick benchmark test result:

Generating 100000 random numbers from 0 to int32.maxvalue with your function:  
15500ms

Generating 100000 random numbers from 0 to int32.maxvalue with Random.Next():  
2ms

**DisguiserNET** - September 1, 2014, 11:51 am

(Sep 1, 2014 07:36 AM)ubbelol Wrote: [\[CITE\]](#) Did some testing just for fun, and to give you a representation of why trying to implement stuff like this yourself is not a good idea I'll show you a quick benchmark test result:

Generating 100000 random numbers from 0 to int32.maxvalue with your function:  
15500ms

Generating 100000 random numbers from 0 to int32.maxvalue with Random.Next():  
2msi had problems with it as well, solved it by initializing the random instance with RNGCryptoServiceProvider

**PapaBeans** - September 1, 2014, 12:04 pm

(Sep 1, 2014 03:35 AM)HeBTMy Wrote: [\[CITE\]](#) Lol, when you're caught stealing, you edit the post saying it was you and your friend, how clever.

Actually when I edited the thread, I labeled it with an EDIT:: tab. It was there from the start

**CottleSilver** - September 1, 2014, 12:51 pm

(Sep 1, 2014 12:04 PM)PapaBeans Wrote: [\[CITE\]](#) Actually when I edited the thread, I labeled it with an EDIT:: tab. It was there from the start

I remember the original thread, you edited the part about 'the friend', stop the bullshit

**Mr.Steven** - September 1, 2014, 12:55 pm

(Sep 1, 2014 12:04 PM)PapaBeans Wrote: [\[CITE\]](#) Actually when I edited the thread, I labeled it with an EDIT:: tab. It was there from the startYou shouldn't lie:

<http://webcache.googleusercontent.com/search?q=cache:CutmjI MAOnAJ:hackforums.net/printthread.php%3Ftid%3D4425355+&cd=1&hl=en&ct=clnk&gl=en>

**Dialog** - September 1, 2014, 1:09 pm

(Sep 1, 2014 12:55 PM)Mr.Steven Wrote: [\[CITE\]](#) You shouldn't lie:

<http://webcache.googleusercontent.com/search?q=cache:CutmjlMAOnAJ:hackforums.net/printthread.php%3Ftid%3D4425355+&cd=1&hl=en&ct=clnk&gl=en>

Ouch. Let's see what OP has to say to this ?

**DragonHunter** - September 1, 2014, 2:57 pm

Like UbbeLol and others said... this is not completely random, it is in some way of course since it's showing you a different number each time you'd maybe call it

But this thing is using too many API's, and if you'd have had some super computer and calling the function twice it could show the same result

The Cursor.Position may even throw a error when using it in a Windows Service, not sure about it

Maybe a fun way of making a random function is to generate a big array from 0 to 100000 and using the Fisher–Yates shuffle algorithm, still not random enough, but fun todo

**Rolex™** - September 1, 2014, 5:22 pm

Nothing is completely random like the others said. I would just use the built in class or if you really wanted to get as random as you could use <http://Random.org>. They use radio receivers to pick up atmospheric noise, which is quite "Random". If you wanted to do it, it's the simplest thing to do.

(Excuse the C# Code)

PHP Code

```
<?php
```

```
using(Webclient client = new Webclient()){
```

```
string RandomNumber = client.DownloadString(
```

```
"http://www.random.org/integers/?num=1&min=1&max=100&col=1&base=10&format=plain&rnd=new");
```

```
int Ran = int.Parse(RandomNumber);
```

```
}
```

If you wanted a little more customization:

PHP Code

```
<?php
```

```
public static int Randomint(int min, int max)
```

```
{
```

```
    using (WebClient client = new WebClient())
```

```
{
```

```
        string RandomNumber = client.DownloadString(
```

```
"http://www.random.org/integers/?num=1&min=" + min + "&max=" + max +
```

```
"&col=1&base=10&format=plain&rnd=new");
```

```
        return int.Parse(RandomNumber);
```

```
    }
```

```
}
```

This would generate a random number between 1 -100, and would be the most efficient way to go about achieving as close to "Random" as possible.

**tweety** - September 1, 2014, 6:32 pm

(Sep 1, 2014 05:22 PM)Rolex™ Wrote: [\[CITE\]](#) Nothing is completely random like the others said. I would just use the built in class or if you really wanted to get as random as you could use <http://Random.org>. They use radio receivers to pick up atmospheric noise, which is quite "Random".

If you wanted to do it, it's the simplest thing to do.

(Excuse the C# Code)

PHP Code

```
<?php
```

```
using(Webclient client = new Webclient()){
```

```
string RandomNumber = client.DownloadString(
```

```
"http://www.random.org/integers/?num=1&min=1&max=100&col=1&base=10&format=plain&rnd=new");
```

```
int Ran = int.Parse(RandomNumber);
```

```
}
```

If you wanted a little more customization:

PHP Code

```
<?php
```

```
public static int Randomint(int min, int max)
```

```
{
```

```
    using (WebClient client = new WebClient())
```

```
{
```

```
    string RandomNumber = client.DownloadString(
```

```
"http://www.random.org/integers/?num=1&min=" + min + "&max=" + max +
```

```
"&col=1&base=10&format=plain&rnd=new");
```

```
    return int.Parse(RandomNumber);
```

```
}
```

```
}
```

This would generate a random number between 1 -100, and would be the most efficient way to go about achieving as close to "Random" as possible.

No offence but this is probably the worst method you can think of, using an external source that relies on an internet connection?

If the site goes down, if no internet connection is found, if the page is moved.. So many risky factors.

**TheBarret** - September 1, 2014, 6:45 pm

Tweet is Chrome trolling me or is the background color of your singature different from the hf bg color o.O

OT: I rather fuck a random hooker with the possibility to get scabbies then using your code.

- Barret

**\$lick** - September 1, 2014, 8:50 pm

(Sep 1, 2014 06:45 PM)TheBarret Wrote: [CITE](#) Tweet is Chrome trolling me or is the background color of your singature different from the hf bg color o.O

OT: I rather fuck a random hooker with the possibility to get scabbies then using your code.

- Barret

Yeah he seems to be using a lighter version.

**Rolex™** - September 2, 2014, 6:11 am

(Sep 1, 2014 06:32 PM)Tweety. Wrote: [\[CITE\]](#) No offence but this is probably the worst method you can think of, using an external source that relies on an internet connection?

If the site goes down, if no internet connection is found, if the page is moved.. So many risky factors.

You're saying it's a bad idea though based off those factors. I'm saying it's the best idea provided that you have internet. Lets be honest I don't think Random.org is going down anytime soon.

I understand what you're saying, that you're relying on that source. It is the best METHOD though for randomness. Unless of course you wanted to duplicate getting randoms through atmospheric noise via VB.NET.

**sh@rp** - September 2, 2014, 10:05 am

lol : caught red handed....

His intention was bad and it would be wise to close this thread !

I say this .....

**ubbelol** - September 4, 2014, 2:38 am

(Sep 2, 2014 06:11 AM)Rolex™ Wrote: [\[CITE\]](#) You're saying it's a bad idea though based off those factors. I'm saying it's the best idea provided that you have internet. Lets be honest I don't think Random.org is going down anytime soon.

I understand what you're saying, that you're relying on that source. It is the best METHOD though for randomness. Unless of course you wanted to duplicate getting randoms through atmospheric noise via VB.NET.

The main issue is you're butchering the performance of a random function. Imagine if you need to quickly generate 1000 random numbers, your method will take quite a bit of time while the built-in random is near instant.

**Bcoder** - September 4, 2014, 2:54 am

i dont understand what you do, but whatever good job :)

**LordCoder** - September 4, 2014, 5:16 am

(Sep 2, 2014 06:11 AM)Rolex™ Wrote: [\[CITE\]](#) You're saying it's a bad idea though based off those factors. I'm saying it's the best idea provided that you have internet. Lets be honest I don't think Random.org is going down anytime soon.

I understand what you're saying, that you're relying on that source. It is the best METHOD though for randomness. Unless of course you wanted to duplicate getting randoms through atmospheric noise via VB.NET.

I still didn't see duplicate randoms by Random class USED properly. Of course it's quite similar because of the seed. Do your jobs and you'll get a ~95% random number. I did one in 5 mins, it's not difficult and no need to rely on a website.