

Forum:[Hack Forums](#)

Thread: [Best way to use a progressbar](#)

AspectModding - March 26, 2016, 11:27 pm

Hello there HF, toady I here asking some advice/help on the best way to use a progress bar. The way I do it now is so terrible, example Progressbar1.value = 5;, Progressbar1.value = 10;, and so on. But I would like to show the correct percentage not a rough guess. I looked on codeproject and msdn but as always they like to make it look very confusing. Could anyone point me in the right path please. :)

Odus - March 27, 2016, 1:59 am

What kind of task are you showing progress for? It varies.

hackpet - March 27, 2016, 3:02 am

If you know the correct percentage, F.e.: if you copy files this can be the finished files/all files (assuming that the files are aproxx same size). Or if you know how long will a specific task take: elapsed time/full time....

Then every time when you finished a part of the job, you just update the value. You can also use the methods: Increment(...) or PerformStep(...).

So i guess, setting the value is a good way, if you know the correct value. Like you did it in the example `Pbar.Value=(int)(100.0*finishedsteps/allsteps);`

If you need help calculating the correct percentage, you must be more specific about the problem.

GuRuHaX - March 27, 2016, 10:25 am

Like Odus says it all depends on what the task is but here a video i used when first using progressBar :

<https://www.youtube.com/watch?v=b5pFzMo2wR0>

Hope it helps !

th3s3cr3t0n3 - March 28, 2016, 12:43 pm

True, depends on what you're doing. But... I've used it before with a webrequest to a page, get the response put it into a stream, get the byte array, then you can set the maximum value equal to the byte array length. Once you do that you could traverse the bytes in a loop or something and increment the progress bar X% when a condition is met, say you might do it only 10% so use mod or do it every byte, up to you. That's an example I've done.

I've done the above but with using an async await tasks and then invoke the background thread to modify the GUI for best results, and so the async await tasks way is only available .NET 4+ I believe, but I know you can get a Nuget package that support async await tasks. Just thought I'd mention, I only do this as I have a few web requests and web service calls every few seconds, so to keep the gui quick I use async.