

گزارش درصد پیشرفت عملیات در اعمال طولانی، [امکان لغو](#) هوشمندانه‌تری را برای کاربر فراهم می‌کند. در دات نت 4.5 دو روش برای گزارش درصد پیشرفت عملیات اعمال غیرهمزمان تدارک دیده شده‌اند:

- اینترفیس جنریک IProgress واقع در فضای نام System

- کلاس جنریک Progress واقع در فضای نام System

در اینجا وهله‌ی از پیاده سازی اینترفیس IProgress به Task ارسال می‌شود. در این بین، عملیات در حال انجام با فراخوانی متد Report آن می‌تواند در هر زمانیکه نیاز باشد، درصد پیشرفت کار را گزارش کند.

```
namespace System
{
    public interface IProgress<in T>
    {
        void Report( T value );
    }
}
```

البته برای اینکه کار تعریف و پیاده سازی اینترفیس IProgress اندکی کاهش یابد، کلاس توکار Progress برای اینکار تدارک دیده شده‌است. نکته‌ی مهم آن استفاده از Synchronization Context برای ارائه گزارش پیشرفت در ترد UI است تا به سادگی بتوان از نتایج دریافتی، در رابط کاربری استفاده کرد.

```
namespace System
{
    public class Progress<T> : IProgress<T>
    {
        public Progress();
        public Progress( Action<T> handler );
        protected virtual void OnReport( T value );
    }
}
```

یک مثال از گزارش درصد پیشرفت عملیات به همراه پشتیبانی از لغو آن

```
using System;
using System.Threading;
using System.Threading.Tasks;

namespace Async09
{
    public class TestProgress
    {
        public async Task DoProcessingReportProgress()
        {
            var progress = new Progress<int>(percent =>
            {
                Console.WriteLine(percent + "%");
            });

            var cts = new CancellationTokenSource();

            // call some where cts.Cancel();

            try
            {
                await doProcessing(progress, cts.Token);
            }
            catch (OperationCanceledException ex)
            {
            }
        }
    }
}
```

```

        //todo: handle cancellations
        Console.WriteLine(ex);
    }

    Console.WriteLine("Done!");
}

private static async Task doProcessing(IProgress<int> progress, CancellationToken ct)
{
    await Task.Run(async () =>
    {
        for (var i = 0; i != 100; ++i)
        {
            await Task.Delay(100, ct);
            if (progress != null)
                progress.Report(i);

            ct.ThrowIfCancellationRequested();
        }
    }, ct);
}
}
}

```

متد `private static async Task doProcessing` طوری طراحی شده است که از مفاهیم لغو یک عملیات غیرهمزمان و همچنین گزارش درصد پیشرفت آن توسط اینترفیس `IProgress` پشتیبانی می کند. در اینجا هر زمانیکه نیاز به گزارش درصد پیشرفت باشد، متد `Report` وهله ای ارسالی به آرگومان `progress` فراخوانی خواهد شد.

برای تدارک این وهله، از کلاس توکار `Progress` دات نت در متد `public async Task DoProcessingReportProgress` استفاده شده است.

این متد جنریک بوده و برای مثال نوع آن در اینجا `int` تعریف شده است. سازنده ای آن می تواند یک `callback` را قبول کند. هر زمانیکه متد `Report` در متد `doProcessing` فراخوانی گردد، این `callback` در سمت کدهای استفاده کننده، فراخوانی خواهد شد. مثلا توسط مقدار آن می توان یک `Progress bar` را نمایش داد.

به علاوه روش دیگری را در مورد لغو یک عملیات در اینجا ملاحظه می کنید. متد `ThrowIfCancellationRequested` نیز سبب خاتمه ای عملیات می گردد؛ البته اگر در کدهای برنامه در جایی متد `Cancel` توکن، فراخوانی گردد. برای مثال یک دکمه ای لغو عملیات در صفحه قرارگیرد و کار آن صرفا فراخوانی `cts.Cancel` باشد.