

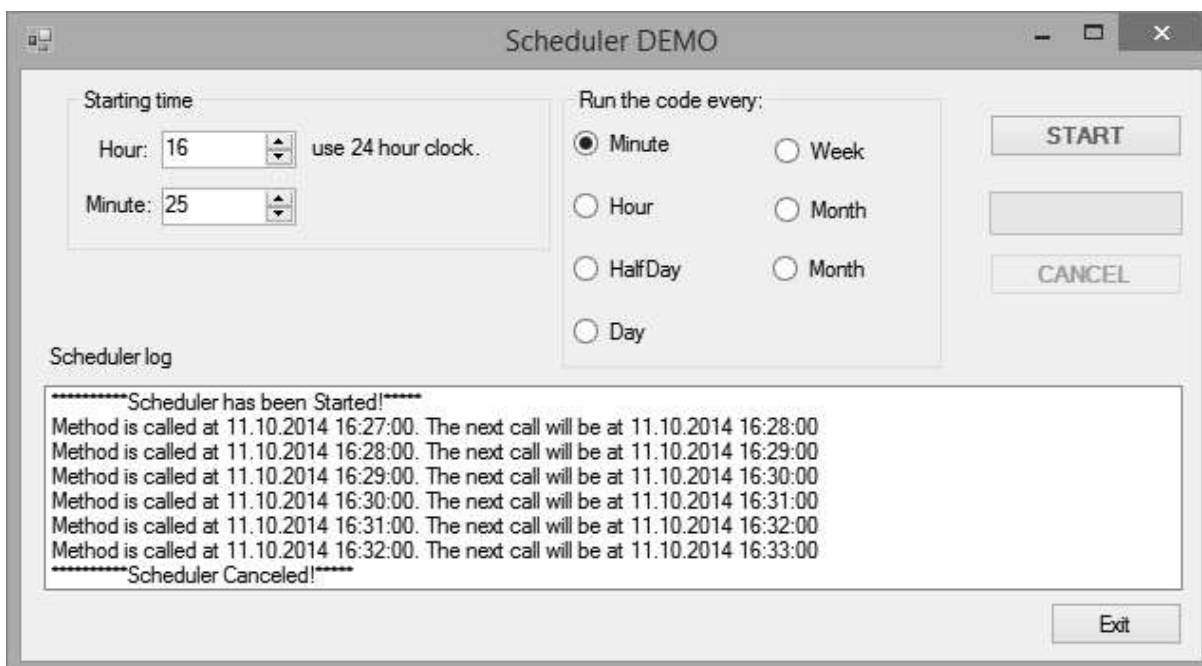


How to Run Code Daily at a Specific Time in C#: Part 2



Bahrudin Hrnjica
7 Jan 2015 CPOL

How to run code daily at a specific time in C#



Few months ago, I wrote a blog post about how to run code daily at a specific time. I didn't know that the post will be the most viewed post on my blog. Also, there were several questions how to implement the complete example. So today, I have decided to write another post, and extend my previous post in order to answer these questions as well as to generalize this subject into a cool demo called Scheduler DEMO.

The post is presenting simple Windows Forms application which calls a method for every minute, day, week, month or year. Also, the demo shows how to cancel the scheduler at any time.

The picture above shows a simple Windows Forms application with two numeric controls in which you can set starting hour and minute for the scheduler. Next, there is a button **Start** to activate timer for running code, as well as **Cancel** button to cancel the scheduler. When the time comes, the application writes the message on the Scheduler Log.

Implementation of the Scheduler

Scheduler is started by clicking the Start button which is implemented with the following code:

C#

```
/// <summary>
/// Setting up time for running the code
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void startBtn_Click(object sender, EventArgs e)
{
    //retrieve hour and minute from the form
    int hour = (int)numHours.Value;
    int minutes = (int)numMins.Value;

    //create next date which we need in order to run the code
    var dateNow = DateTime.Now;
    var date = new DateTime(dateNow.Year, dateNow.Month, dateNow.Day, hour, minutes, 0);

    listBox1.Items.Add("*****Scheduler has been started!*****");

    //get nex date the code need to run
    var nextDateValue=getNextDate(date,getScheduler());

    runCodeAt(nextDateValue, getScheduler());
}
```

When the time is defined, then the runCodeAt method is called which implementation can be like the following:

C#

```
/// <summary>
/// Determine the timeSpan Delay must wait before running the code
/// </summary>
/// <param name="date"></param>
/// <param name="scheduler"></param>
private void runCodeAt(DateTime date,Scheduler scheduler )
{
    m_ctSource = new CancellationTokenSource();

    var dateNow = DateTime.Now;
    TimeSpan ts;
    if (date > dateNow)
        ts = date - dateNow;
    else
    {
        date = getNextDate(date, scheduler);
        ts = date - dateNow;
    }

    //enable the progressbar
    prepareControlForStart();

    //waits for a certain time and runs the code,
    //in the mean time, you can cancel the task at any time
    Task.Delay(ts).ContinueWith((x) =>
    {
        //run the code at the time
        methodToCall(date);
    });
}
```

```
        //setup call next day
        runCodeAt(getNextDate(date, scheduler), scheduler);
    },m_ctSource.Token);
}
```

The method above creates the `cancellationToken` needed for cancel the scheduler, calculate `timeSpan` – total waiting time, then when the time comes, call the method `methodToCall` and calculate the next time for running the scheduler.

This demo also shows how to wait for a certain amount of time without blocking the UI thread.

The full demo code can be found on [OneDrive](#).

License

This article, along with any associated source code and files, is licensed under [The Code Project Open License \(CPOl\)](#)

About the Author



Bahrudin Hrnjica



Software Developer (Senior)
Bosnia and Herzegovina 🇧🇦


Bahrudin Hrnjica holds a Ph.D. degree in Technical Science/Engineering from University in Bihać.

Besides teaching at University, he is in the software industry for more than two decades, focusing on development technologies e.g. .NET, Visual Studio, Desktop/Web/Cloud solutions.

He works on the development and application of different ML algorithms. In the development of ML-oriented solutions and modeling, he has more than 10 years of experience. His field of interest is also the development of predictive models with the ML.NET and Keras, but also actively develop two ML-based .NET open source projects: GPdotNET-genetic programming tool and ANNdotNET - deep learning tool on .NET platform. He works in multidisciplinary teams with the mission of optimizing and selecting the ML algorithms to build ML models.

He is the author of several books, and many online articles, writes a blog at <http://bhrnjica.net>, regularly holds lectures at local and regional conferences, User groups and Code Camp gatherings, and is also the founder of the Bihac Developer Meetup Group. Microsoft recognizes his work and awarded him with the prestigious Microsoft MVP title for the first time in 2011, which he still holds today.

Comments and Discussions

 **9 messages** have been posted for this article Visit <https://www.codeproject.com/Articles/828791/How-to-Run-Code-Daily-at-a-Specific-Time-in-Csharp> to post and view comments on this article, or click [here](#) to get a print view with messages.

[Permalink](#)
[Advertise](#)
[Privacy](#)
[Cookies](#)
[Terms of Use](#)

Article Copyright 2014 by Bahrudin Hrnjica
Everything else Copyright © CodeProject,
1999-2022

Web04 2.8.2022.07.15.1