Sql server, .net and c# video tutorial

Free C#, .Net and Sql server video tutorial for beginners and intermediate programmers.

BackgroundWorker Class example in windows forms application

In this video, we will discuss the use of **BackgroundWorker** Class in windows forms application with an example. Along the way, we will also discuss using **progressbar** control.

First, let's understand, why should we use BackgroundWorker Class?
By default windows applications are single threaded. This means, when we run the application there will be a single thread which is commonly called as UI thread. This is the thread that is responsible for doing all the work, that is
1. Creating and updating user interface elements and
2. Executing application code

For example, let's say we have a function that takes 10 seconds to complete.
While the UI thread is busy processing this function, the UI of the application is frozen and the end user cannot interact with the controls on the form during those 10 seconds.

Complete Tutorials

Angular tutorial for beginners

Angular 5 Tutorial for beginners

Important Videos

The Gift of Education

Web application for your business

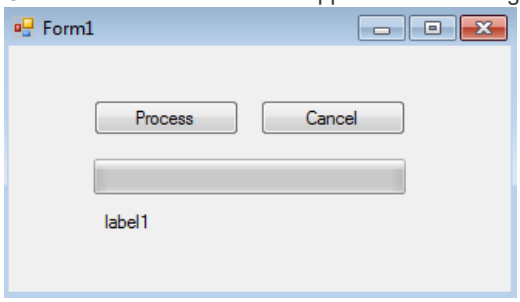
How to become .NET developer

Resources available to help you

To keep our application responsive, we can create a new Thread ourselves and then ask that thread to execute the function that takes long time to complete, so that the UI thread is free to update the controls on the form. But the problem with creating threads ourselves is that, it can get extremely complex especially when we have to update UI controls based on the status of the thread function execution. It's dangerous to access UI objects on a thread that didn't create them. So the better option here is to use, BackgroundWorker Class which simplifies multithreading.

Using BackgroundWorker Class

Create a new windows forms application and design the form as shown below.



The windows form contains

a) Two Buttons
b) Progressbar control
c) Label control

Now, drag and drop **BackgroundWorker** object from toolbox onto the form.

On Button1 set the following properties
Name = btnProcess
Text = Process

On Button2 set the following properties
Name = btnCancel
Text = Cancel

Properties of BackgroundWorker class

WorkerSupportsCancellation - Set this property to true if you want the background operation to allow cancellation.
WorkerReportsProgress - Set this property to true if you want the background operation to report progress.

Events of BackgroundWorker class

DoWork - The DoWork event handler is where the time consuming operation runs on the background thread.

```
private void backgroundWorker1_DoWork(object sender, DoWorkEventArgs e)
{
    int sum = 0;
    for (int i = 1; i <= 100; i++)
    {
        Thread.Sleep(100);
        sum = sum + i;
        // Calling ReportProgress() method raises ProgressChanged event
        // To this method pass the percentage of processing that is complete
        backgroundWorker1.ReportProgress(i);

        // Check if the cancellation is requested
        if (backgroundWorker1.CancellationPending)
        {
            // Set Cancel property of DoWorkEventArgs object to true
            e.Cancel = true;
            // Reset progress percentage to ZERO and return
```

Dot Net Video Tutorials

ASP.NET Core Tutorial
Angular 6 Tutorial
Angular CRUD Tutorial
Angular CLI Tutorial
Angular 2 Tutorial
Design Patterns
SOLID Principles
ASP.NET Web API
Bootstrap
AngularJS Tutorial
jQuery Tutorial
JavaScript with ASP.NET Tutorial
JavaScript Tutorial
Charts Tutorial
LINQ
LINQ to SQL
LINQ to XML
Entity Framework
WCF
ASP.NET Web Services
Dot Net Basics
C#
SQL Server
ADO.NET
ASP.NET
GridView
ASP.NET MVC
Visual Studio Tips and Tricks
Dot Net Interview Questions

Slides

Entity Framework
WCF
ASP.NET Web Services
Dot Net Basics
C#

```
            backgroundWorker1.ReportProgress(0);
            return;
        }
    }

    // Store the result in Result property of DoWorkEventArgs object
    e.Result = sum;

}
```

ProgressChanged - The ProgressChanged event handler is where we write code to update the user interface elements with the progress made so far.

```
private void backgroundWorker1_ProgressChanged
    (object sender, ProgressChangedEventArgs e)
{
    progressBar1.Value = e.ProgressPercentage;
    label1.Text = e.ProgressPercentage.ToString() + "%";

}
```

RunWorkerCompleted - This event gets raised for 3 reasons
a) When the background worker has completed processing successfully
b) When the background worker encountered an error
c) When the background worker is requested to cancel the execution

```
private void backgroundWorker1_RunWorkerCompleted
    (object sender, RunWorkerCompletedEventArgs e)
{
    if (e.Cancelled)
    {
        label1.Text = "Processing cancelled";
    }
    else if (e.Error != null)
    {
        label1.Text = e.Error.Message;
    }
    else
    {
        label1.Text = e.Result.ToString();
    }
}

private void btnProcess_Click(object sender, EventArgs e)
{
    // Check if the backgroundWorker is already busy running the asynchronous
operation
    if (!backgroundWorker1.IsBusy)
    {
        // This method will start the execution asynchronously in the background
        backgroundWorker1.RunWorkerAsync();
    }
}

private void btnCancel_Click(object sender, EventArgs e)
{
    if (backgroundWorker1.IsBusy)
    {
        // Cancel the asynchronous operation if still in progress
        backgroundWorker1.CancelAsync();
    }

}
```

SQL Server

ADO.NET

ASP.NET

GridView

ASP.NET MVC

Visual Studio Tips and Tricks

Java Video Tutorials

Part 1 : Video | Text | Slides

Part 2 : Video | Text | Slides

Part 3 : Video | Text | Slides

Interview Questions

C#

SQL Server

Written Test

4 comments:

RaaJ March 30, 2014 at 1:18 AM

please provide wpf tutorilas.....

Reply

shaiz AWAN March 30, 2014 at 8:32 PM

Hy dear vankat thank you so much, that you have start to windows programing now is it start to wpf?

Reply

Unknown May 10, 2017 at 9:00 AM

Nice....
Can we do the same in python.

Reply

Ariful Islam (Arif) May 15, 2018 at 10:34 AM

Thank you very much for making it easy to understand.

Reply

Enter your comment...

Comment as: Hiện Doãn ((▼ Sign out

Publish Preview Notify me

If you like this website, please share with your friends on facebook and Google+ and recommend us on google using the g+1 button on the top right hand corner.

Links to this post

Create a Link

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)