

## Re: Executing a program while displaying a model window

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2004-03/2123.html>

---

**From:** Ignacio Machin \(.NET/ C# MVP\) ("*Ignacio*")

**Date:** 03/08/04

Date: Mon, 8 Mar 2004 14:11:02 -0500

Hi Erick,

I do a similar thing, What I do is that the main thread is used for the UI, I display the form with ShowDialog() and is inside this form where I create the working thread where I perform the works:

I do something little different though, I'm copying a group of files from the desktop to a PocketPC so my working thread perform the copy and each time a file is copied the progress bar is updated with the name of the file just copied. but other than that it does what you are trying to do.

Below you will find the code I use:

```
//Open the form ( from main form )
private void button2_Click(object sender, System.EventArgs e)

{

new ExportImages().ShowDialog();

}

In ExportImages form:
public delegate void ProcessedFileHandler( ); // I use this delegate to send
an event from the working thread to the UI thread.

//Create the thread and init the progress bar

private void ImportImages_Load(object sender, System.EventArgs e)

{

this.files = Directory.GetFiles( Config.ExportImageDirectory);

this.progressBar1.Minimum = 0;
```

microsoft.public.dotnet.languages.csharp: Re: Executing a program while displaying a model window

```
this.progressBar1.Maximum = files.Length;

procfilehandler = new ProcessedFileHandler( this.UpdateProgressBar);

//Start the thread

this.workingThread = new Thread( new ThreadStart( this.ExportImage ));

this.workingThread.Start();

}

//The working thread method., take a look how the event is invoked on the
thread that the control belong to:
void ExportImage( )
{
    foreach(string imageURL in files)
    {
        this.currentpos++;
        this.currentfile = Path.GetFileName(imageURL);

        //Update the interface
        this.progressBar1.Invoke( procfilehandler, null);

        //Use RAPI to copy the file to the PPC
        rapi.CopyFileToDevice( imageUrl, targeurl , true);
    }
    //All we need is a control in the main thread to execute a method on the
main thread.
    this.progressBar1.Invoke( new ProcessedFileHandler( this.Done), null);
}

//Here is the Handler of the Update Progress bar.
void UpdateProgressBar( )
{
    this.progressBar1.Value++;
    FileNameLBL.Text = currentfile;
    currentLBL.Text = currentpos.ToString();
}
```

Hope this help,

```
--
Ignacio Machin,
ignacio.machin AT dot.state.fl.us
Florida Department Of Transportation
"Eric Paul" <ericallenpaul@hotmail.com> wrote in message
news:658d9a02.0403081019.7d773d0e@posting.google.com...
> I have searched in vain for what seems like something that should be
> simple to accomplish.
>
> I have a Windows application written in C# that scans a driver's
> license. While it's scanning I want to display a "Scanning..."
```

Re: Executing a program while displaying a model window

## microsoft.public.dotnet.languages.csharp: Re: Executing a program while displaying a model window

```
> progress bar window until the scanning is complete or the user clicks
> cancel.
> All is fine as long as I use "form.visible" to display the window with
> the progress bar.
> I use another thread to display the window and then I kill the thread
> when the scanning completes.
> The problem is that I want this "progress bar" window to be modal--not
> modeless. If I use the "form.ShowDialog()" all processing stops in the
> new thread--the progress bar never updates until the process is
> complete.
>
> Is there a way to use a modal window and still continue program
> execution?
>
> I have searched and found some references to using "Application.Run()"
> and "Form.Invoke" but they're not really very clear on how to
> accomplish this. It seems like this problem should have been solved
> before since many applications have modal "progress bar" windows but I
> can't find any good examples. Here is a sampling of my current code:
>
>
> Thread t = new Thread(new ThreadStart(ThreadProc));
> t.Start();
>
> myScan.Scan();
> Byte[] byteBLOBData = new Byte[0];
> FileStream fs = new
FileStream(ConfigurationSettings.AppSettings["ImageFilename"],
> FileMode.Open, FileAccess.Read);
> photoBox1.Image= Image.FromStream(fs);
> fs.Close();
> myScan.ExtractOCRData();
> t.Abort();
> myScan.ExtractFaceImage();
> myScan.ExtractSignature();
>
> if (myScan.ScannerError == false)
> {
>     txtNameOnCard_OCR.Text = myScan.NameOnCard;
> txtFirstName_OCR.Text = myScan.FirstName;
> txtMiddleName_OCR.Text = myScan.MiddleName;
> txtLastName_OCR.Text = myScan.LastName;
> txtSuffixName_OCR.Text = myScan.SuffixName;
> txtAddress_OCR.Text = myScan.Address;
> txtCity_OCR.Text = myScan.City;
> txtState_OCR.Text = myScan.State;
> txtZip_OCR.Text = myScan.Zip;
> txtDOB_OCR.Text = myScan.DOB;
> txtIssue_OCR.Text = myScan.Issue;
> txtExpires_OCR.Text = myScan.Expires;
> txtID_OCR.Text = myScan.ID;
> txtAccuracy_OCR.Text = myScan.Accuracy + "";
> }
> else
> {
>     MessageBox.Show(myScan.ScannerErrorMessage);
> myScan.ScannerError = false;
> return;
> }
>
>
> public static void ThreadProc()
```

## microsoft.public.dotnet.languages.csharp: Re: Executing a program while displaying a model window

```
> {
> // construct a new progress dialog
> frmProgress myProgressDialog = new frmProgress();
>
> myProgressDialog.progressBar1.Minimum = 0;
> myProgressDialog.progressBar1.Maximum = 9;
> myProgressDialog.progressBar1.Step = 1;
> myProgressDialog.progressBar1.Value = 0;
>
> myProgressDialog.FormBorderStyle = FormBorderStyle.FixedDialog;
> myProgressDialog.StartPosition = FormStartPosition.CenterScreen;
> myProgressDialog.Visible = true;
> myProgressDialog.Update();
> myProgressDialog.Refresh();
> myProgressDialog.CancelButton = myProgressDialog.btnCancel;
>
> while ( myProgressDialog.progressBar1.Value <
> myProgressDialog.progressBar1.Maximum )
> {
> myProgressDialog.progressBar1.Value++;
> System.Threading.Thread.Sleep(1000);
> myProgressDialog.Update();
> myProgressDialog.Refresh();
> }
>
> //System.Threading.Thread.Sleep(2500);
> myProgressDialog.Visible = false;
> }
>
> Any and all suggestions are most welcome.
>
> Thanks,
>
> Eric
```