

The left side of the slide features a vertical stack of decorative elements. At the top, there are several thin, light gray vertical bars of varying widths. Below these, a series of dark blue circles of different sizes are arranged in a descending, staggered pattern. The largest circle is at the top left, with smaller circles positioned to its right and below it, creating a sense of movement or a trail.

# KEEP YOUR UI RESPONSIVE WITH THE BACKGROUNDWORKER COMPONENT

Presented by **Jeremy Clark**  
[www.jeremybytes.com](http://www.jeremybytes.com)

# AGENDA

- About Me
- BackgroundWorker Component Features
- Overall Workflow
- LOOK AT THE CODE
- Resources



## ABOUT ME

- Jeremy Clark
- Work for **BITKOO**<sup>®</sup>
- 10 Years as an Application Developer
- .NET/C# since 2005
- ASP.NET, WinForms, Web Services, WCF, WPF, Silverlight, ASP.NET MVC
- [www.jeremybytes.com](http://www.jeremybytes.com)



Jeremy – 7<sup>th</sup> Grade



WWW.JEREMYBYTES.COM



# BACKGROUNDWORKER COMPONENT FEATURES

- Background Thread Handling
- Automatic Marshalling Across Threads
- Progress Notification
- Error Handling
- Cancellation

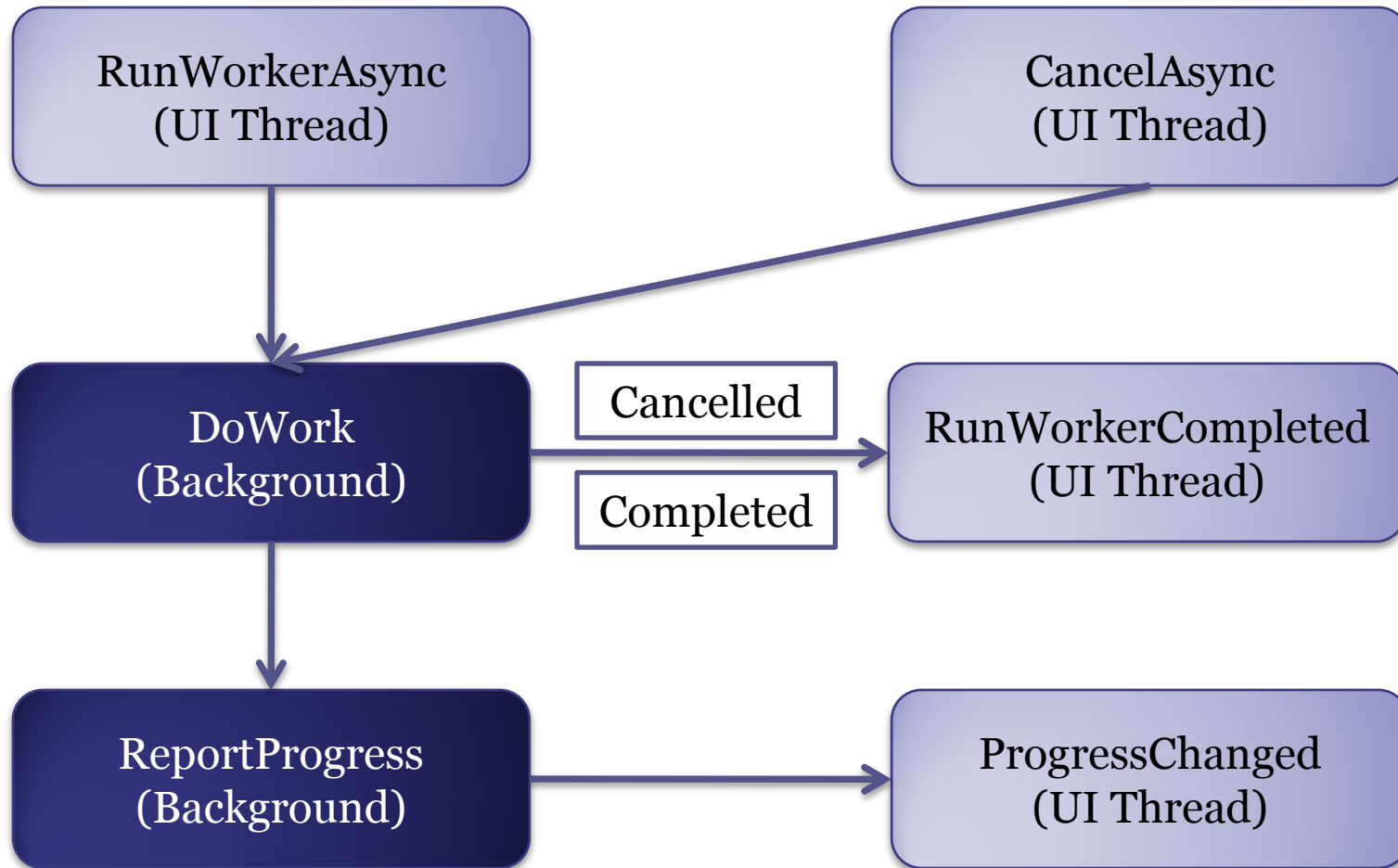


# BACKGROUNDWORKER MEMBERS

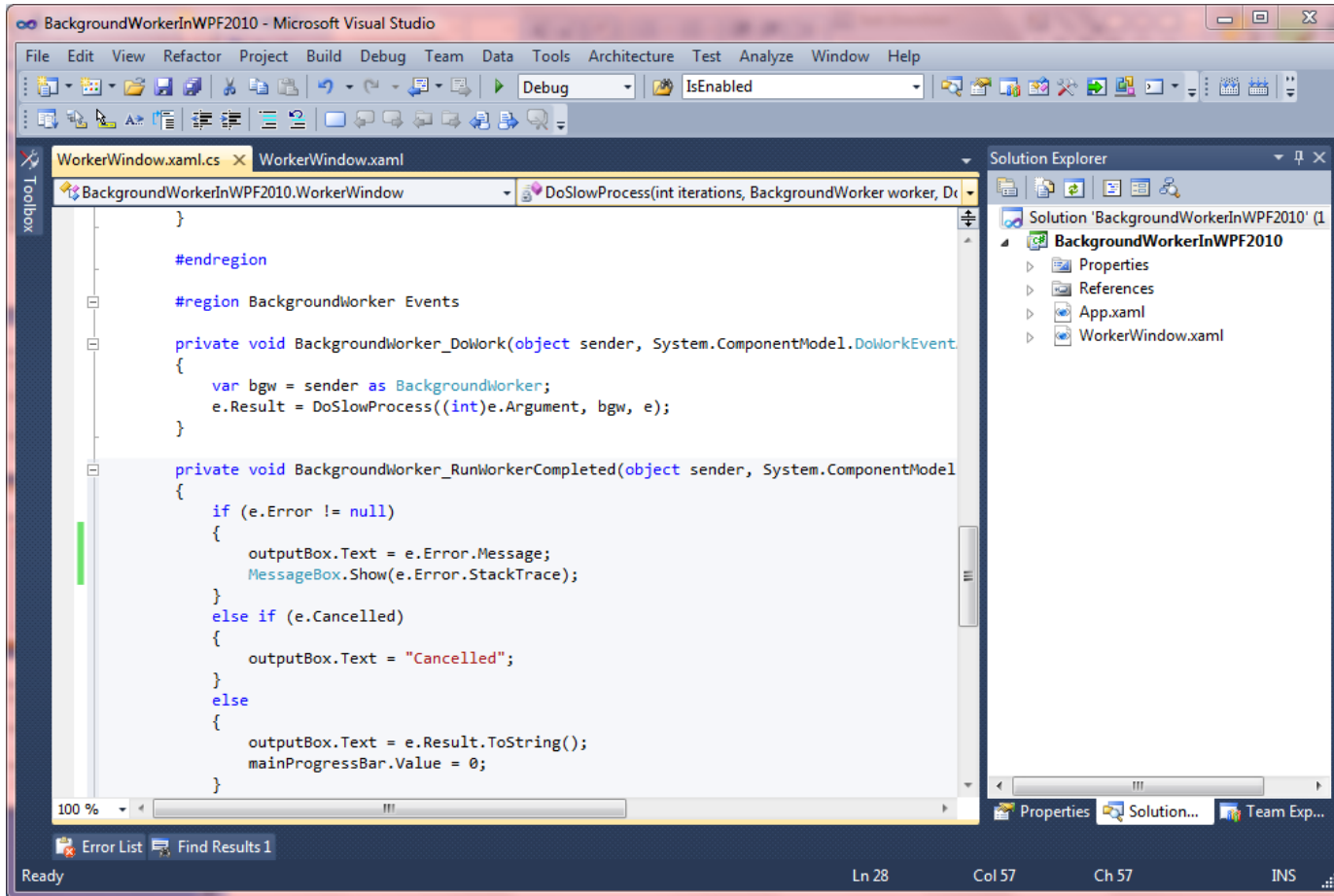
- Methods
  - RunWorkerAsync
  - ReportProgress
  - CancelAsync
- Properties
  - WorkerReportsProgress
  - WorkerSupportsCancellation
  - CancellationPending
  - IsBusy
- Events
  - DoWork
  - RunWorkerCompleted
  - ProgressChanged



# OVERALL WORKFLOW



# LOOK AT THE CODE



```

}

#endregion

#region BackgroundWorker Events

private void BackgroundWorker_DoWork(object sender, System.ComponentModel.DoWorkEventArgs e)
{
    var bgw = sender as BackgroundWorker;
    e.Result = DoSlowProcess((int)e.Argument, bgw, e);
}

private void BackgroundWorker_RunWorkerCompleted(object sender, System.ComponentModel.RunWorkerCompletedEventArgs e)
{
    if (e.Error != null)
    {
        outputBox.Text = e.Error.Message;
        MessageBox.Show(e.Error.StackTrace);
    }
    else if (e.Cancelled)
    {
        outputBox.Text = "Cancelled";
    }
    else
    {
        outputBox.Text = e.Result.ToString();
        mainProgressBar.Value = 0;
    }
}

```





# RESOURCES

- Pro WPF in C# 2010 by Matthew MacDonald
  - <http://www.apress.com/book/view/1430272058>
  - Contains an overview of the BackgroundWorker and a WPF example (prime number calculation)
- MSDN – BackgroundWorker Component
  - <http://msdn.microsoft.com/en-us/library/c8dcext2.aspx>
  - Contains an overview of the BackgroundWorker and a WinForms example (Fibonacci sequence)
- JeremyBytes
  - <http://www.jeremybytes.com/Demos.aspx>
  - Contains downloadable code and full walkthrough
  - Email: [jeremy@jeremybytes.com](mailto:jeremy@jeremybytes.com)

