

```
447 static void Main()
448 {
449     Application.EnableVisualStyles();
450     Application.DoEvents();
451     Application.Run(new MainForm());
452 }
453
454 private void startButton_Click(object sender, System.EventArgs e)
455 {
456     if (runningTask == null)
457     {
458         try
459         {
460             stopButton.Enabled = true;
461             startButton.Enabled = false;
462
463             // Create a new task
464             myTask = new Task();
465
466             // Create a virtual channel
467             myTask.AIChannels.CreateVoltageChannel(physicalChannelComboBox.Text, "",
468             (AITerminalConfiguration)(-1), Convert.ToDouble(minimumValueNumeric.Value),
469             Convert.ToDouble(maximumValueNumeric.Value), AIVoltageUnits.Volts);
470
471             // Configure the timing parameters
472             myTask.Timing.ConfigureSampleClock("", Convert.ToDouble(rateNumeric.Value),
473             SampleClockActiveEdge.Rising, SampleQuantityMode.ContinuousSamples, 1000);
474
475             // Verify the Task
476             myTask.Control(TaskAction.Verify);
477
478             // Prepare the table for Data
479             InitializeDataTable(myTask.AIChannels, ref dataTable);
480             acquisitionDataGrid.DataSource = dataTable;
481
482             runningTask = myTask;
483             analogInReader = new AnalogMultiChannelReader(myTask.Stream);
484             analogCallback = new AsyncCallback(AnalogInCallback);
485
486             // Use SynchronizeCallbacks to specify that the object
487             // marshals callbacks across threads appropriately.
488             analogInReader.SynchronizeCallbacks = true;
489             analogInReader.BeginReadWaveform(Convert.ToInt32(samplesPerChannelNumeric.Value),
490             analogCallback, myTask);
491         }
492         catch (DaqException exception)
493         {
494             // Display Errors
495             MessageBox.Show(exception.Message);
496             runningTask = null;
497             myTask.Dispose();
498             stopButton.Enabled = false;
499             startButton.Enabled = true;
500         }
501     }
502 }
503
504 private void AnalogInCallback(IAsyncResult ar)
505 {
506     try
507     {
508         if (runningTask != null && runningTask == ar.AsyncState)
509         {
510             // Read the available data from the channels
511             data = analogInReader.EndReadWaveform(ar);
512
513             // Plot your data here
514             dataToDataTable(data, ref dataTable);
515
516             analogInReader.BeginMemoryOptimizedReadWaveform(Convert.ToInt32
(samplesPerChannelNumeric.Value),
analogCallback, myTask, data);
517         }
518     }
519 }
```

```

520         catch (DaqException exception)
521         {
522             // Display Errors
523             MessageBox.Show(exception.Message);
524             runningTask = null;
525             myTask.Dispose();
526             stopButton.Enabled = false;
527             startButton.Enabled = true;
528         }
529     }
530
531     private void stopButton_Click(object sender, System.EventArgs e)
532     {
533         if (runningTask != null)
534         {
535             // Dispose of the task
536             runningTask = null;
537             myTask.Dispose();
538             stopButton.Enabled = false;
539             startButton.Enabled = true;
540         }
541     }
542
543     private void dataToDataTable(AnalogWaveform<double>[] sourceArray, ref DataTable dataTable)
544     {
545         // Iterate over channels
546         int currentLineIndex = 0;
547         foreach (AnalogWaveform<double> waveform in sourceArray)
548         {
549             for (int sample = 0; sample < waveform.Samples.Count; ++sample)
550             {
551                 if (sample == 10)
552                     break;
553
554                 dataTable.Rows[sample][currentLineIndex] = waveform.Samples[sample].Value;
555             }
556             currentLineIndex++;
557         }
558     }
559
560     public void InitializeDataTable(AIChannelCollection channelCollection, ref DataTable data)
561     {
562         int numOfChannels = channelCollection.Count;
563         data.Rows.Clear();
564         data.Columns.Clear();
565         dataColumn = new DataColumn[numOfChannels];
566         int numOfWorks = 10;
567
568         for (int currentChannelIndex = 0; currentChannelIndex < numOfWorks; currentChannelIndex+
569 +)
570         {
571             dataColumn[currentChannelIndex] = new DataColumn();
572             dataColumn[currentChannelIndex].DataType = typeof(double);
573             dataColumn[currentChannelIndex].ColumnName = channelCollection[currentChannelIndex].
PhysicalName;
574         }
575
576         data.Columns.AddRange(dataColumn);
577
578         for (int currentDataIndex = 0; currentDataIndex < numOfWorks; currentDataIndex++)
579         {
580             object[] rowArr = new object[numOfChannels];
581             data.Rows.Add(rowArr);
582         }

```