Using DATAQ .NET Controls in MATLAB®

Please follow this tutorial step by step to learn how to port previously recorded data from a .wdq file into a MATLAB® matrix with the WinDaq File Format .Net class.

Step 1: Download and Install the DATAQ SDK .Net Class

The DATAQ SDK .Net Class is installed automatically with most WinDaq installations. To check if you already have it installed, look in your WinDaq installation directory (by default C:\DATAQ) for a folder called DATAQ Instruments SDK Beta. This folder contains example programs and documentation for the DATAQ Hardware Control .Net Library, WinDaq File Format .Net Library, and the Common DATAQ .Net Library. This tutorial explains how to use the WinDaq File Format .Net Control with MATLAB®, but there are also examples of how to use the SDK Hardware control with MATLAB®.

The latest version of the DATAQ .Net Controls can be downloaded for free at: http://www.dataq.com/support/downloads/DATAQ Instruments SDK Beta Web Setup.exe

Step 2: Download the WWBExample.m example program

Download the MATLAB® example programs at: http://www.dataq.com/resources/repository/DataqMatlabSDKExamples.zip

Zip file contents:

> **DataqSDKexample145.m** – Main M-File function. This sample code shows how to use the DataqSDK .Net control with a DI-145 Starter kit.

> **DataqSDKexample149.m** – Main M-File function. This sample code shows how to use the DataqSDK .Net control with a DI-149 Starter kit.

> **DataqSDKexample155.m** – Main M-File function. This sample code shows how to use the DataqSDK .Net control with a DI-155 Starter kit.

> **DataqSDKexample245.m** – Main M-File function. This sample code shows how to use the DataqSDK .Net control with a DI-245 Starter kit.

> **DInewdata.m** – Support M-File function. Event listener for the NewData event for the DataqSDK examples. This file is needed in the same folder when using any of the DataqSDK example M-Files.

> **WWBExample.m** – Main M-File function. This is the sample code demonstrated in this tutorial. This example ports previously recorded data from a .wdq file into a MATLAB® matrix.

The WWBExample is not hardware specific and can open a .wdq file generated by any DATAQ Hardware device. If you need help getting started with the SDK Control for hardware that is not

---

listed in the examples, please post a support ticket at: http://www.dataq.com/support/live-agent/

Step 3: Add the directory where the M-files are located to the MATLAB® path.

Use the addpath function from the MATLAB® command prompt. For example, if you unzipped the file to the C:\ drive, type:

```
>> addpath('C\DATAQMATLABNetExamples')
```

Step 4: Run the example program.

Note: ReadDaqFileAnalysis.m uses the WinDaq file C:\DATAQ\Sample.wdq by default. If you do not have the SAMPLE.wdq file in the DATAQ folder on your C drive, the example program will generate a file error until you change the file name in the code.

To run the example program, type the M-file name at the MATLAB® command prompt:

```
>> WWBExample
```

The program should display the file name and sample rate in the Command Window, give the file start date and file end date, and display 10 rows of data starting at file position 1.032:

```
C:\DATAQ\SAMPLE.WDQ
        1000

  DateTime with properties:

            Date: [1x1 System.DateTime]
             Day: 16
       DayOfWeek: Wednesday
       DayOfYear: 320
            Hour: 22
            Kind: Utc
     Millisecond: 0
          Minute: 59
           Month: 11
             Now: [1x1 System.DateTime]
          UtcNow: [1x1 System.DateTime]
          Second: 31
           Ticks: 629206235710000000
       TimeOfDay: [1x1 System.TimeSpan]
           Today: [1x1 System.DateTime]
            Year: 1994
        MinValue: [1x1 System.DateTime]
        MaxValue: [1x1 System.DateTime]

  DateTime with properties:

            Date: [1x1 System.DateTime]
```

Product Links: Data Acquisition | Data Logger

```
         Day:  16
   DayOfWeek:  Wednesday
   DayOfYear:  320
        Hour:  22
        Kind:  Utc
 Millisecond:  0
      Minute:  59
       Month:  11
         Now:  [1x1 System.DateTime]
     UtcNow:   [1x1 System.DateTime]
      Second:  51
       Ticks:  629206235910000000
   TimeOfDay:  [1x1 System.TimeSpan]
       Today:  [1x1 System.DateTime]
        Year:  1994
    MinValue:  [1x1 System.DateTime]
    MaxValue:  [1x1 System.DateTime]

    2.7661    -0.9326    -0.6714    0.0024
    2.9150    -0.8374    -0.6812    0.0220
    3.0591    -0.6909    -0.6812    0.0171
    3.1934    -0.5640    -0.6714    0.0171
    3.3130    -0.4980    -0.6689    0.0537
    3.4180    -0.4810    -0.6836    0.1099
    3.5181    -0.4858    -0.6982    0.1685
    3.6084    -0.4858    -0.7080    0.2246
    3.6914    -0.4688    -0.7104    0.2637
    3.7671    -0.4395    -0.7129    0.2808
```
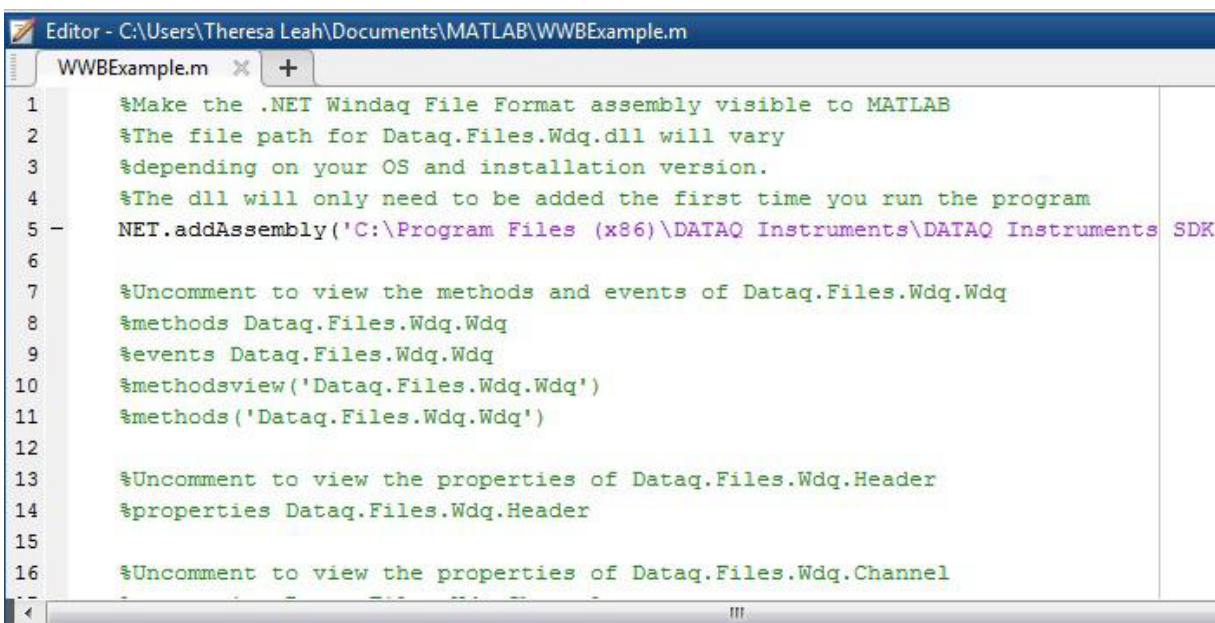
Step 5: Look at the MATLAB® code.

Now that you have the example running in MATLAB®, let's take a look at the code syntax to see what happened when you executed WWBExample:

Open the WWBExample.m file from MATLAB® by going to File>>Open. Select the file WWBExample.m from the location you unzipped it to and click Open. This will open the M-file in the MAT-

LAB® Editor:

Note: The % character designates a comment in MATLAB®.

Let's take a look at the code line by line:

Line 5
```
NET.addAssembly('C:\Program Files (x86)\DATAQ Instruments\DATAQ Instru-
ments SDK Beta\WinDaq File Format .Net library - v2\Da-
taq.Files.Wdq.dll');
```

Makes .NET WinDaq File Format assembly visible to MATLAB®. This dll will only need to be added the first time you run the program. If you get an error, check the file path of the dll, this path will be different depending on your operating system.

Line 19
```
WWBFileName = 'C:\DATAQ\SAMPLE.WDQ';
```

This is the file path of the .wdq file that you want to import. Change this line to the file path that you want to open. If you are opening a network file, make sure that you have full permission and the location mapped from your local computer or the dll may not be able to access the file.

Line 23
```
[wdqRetVal,wdqReadFile] = Dataq.Files.Wdq.Wdq.OpenRead(WWBFileName)
```

Open the .wdq file into the wdqReadFile object. Once this step is complete you will be able to access you file data with the wdqReadFile control. You can use the tab key to display the available

control options from the MATLAB® Command Window.
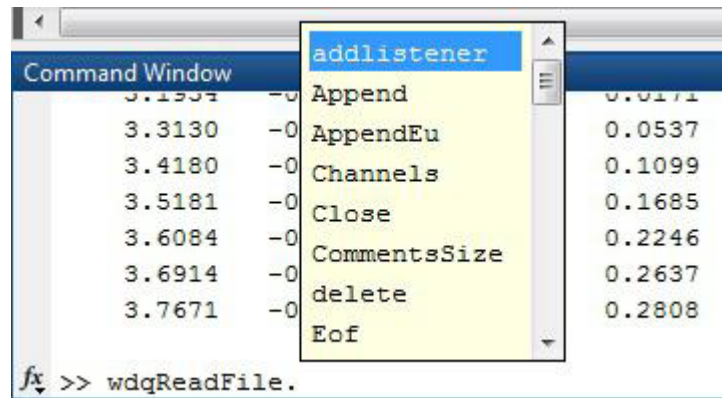
Line 29-32
```
disp(WWBFileName);
disp(wdqReadFile.Header.SampleRate);
disp(wdqReadFile.Header.TimeStarted);
disp(wdqReadFile.Header.TimeStopped);
```

Display some file information – the example program displays the File Name, Sample Rate, Start Date and Time, and End Date and Time of the file.

Line 35-36
```
numSamples = wdqReadFile.Header.DataSize_ / 2;
WWBChannels = wdqReadFile.Header.Channels; %Number of Channels
```

Gets the number of samples in the file and the number of channels. You will need to know the total number of samples if you are going to import the entire file (uncomment line 41). The number of channels is needed to calculate the size of your data import.

Line 39
```
WWB_Rows = 10;
```

Set the number of rows to import. The example program imports 10 rows of data.

Line 44
```
WWBSamplesSelected = WWB_Rows * double(WWBChannels);
```

Get the number of samples to import. Rows * Channels

Line 50
```
WWBFilePosition = 1.032;
```

Set the file position where you want to start importing. WWBFilePosition correlates with WinDaq Playback software's TBF value (Time to Beginning of File) in seconds as displayed in the lower left of the software's display. Press "F3" to enable if the display shows "off." The example program sets the start position at 1.032 which is 5.2% to the EOF. If you move the cursor around in the SAMPLE.wdq file you can see the TBF value change and determine where you would like to start your import.

Line 53-55
```
WWBTimePointer = 1 / wdqReadFile.Header.SampleRate;
wdqFileOffset = WWBFilePosition / WWBTimePointer;
wdqFileSeekOrigin = System.IO.SeekOrigin.Begin;
```

These lines are needed to calculate the start position for the Seek method (**Line 59**).

Line 67
```
WWB_NArray = NET.createArray('System.Double',WWBSamplesSelected);
```

Data coming from the .NET WinDaq File Format control first needs to go into a .NET array (**Line 71**) of System.Double type before it can be written into a MATLAB® matrix. If you attempt to put the data directly into a MATLAB® matrix you will get an error about the data type.

Line 81-92
```
nDataPoint = 1;
for nRows = 1:WWB_Rows
    for nChns = 1:WWBChannels
        try
            %Fill each data point into the matrix WWB_FileData
            WWB_FileData(nRows,nChns)=WWB_NArray(nDataPoint);
        catch ME
            disp(ME.message);
        end
        nDataPoint = nDataPoint + 1;
    end
end
```

Loop through the .NET array and fill the data into the WWB_FileData matrix.

Line 95
```
disp(WWB_FileData)
```

Display the rows of data from the WWB_FileData matix in the Command Window

Step 6: Perform MATLAB® Calculations

Now that you understand the example program code, let's add more code to perform calculations on the WWB_FileData matrix.

Find the Max, Min and Mean

Max:

```
>> max(WWB_FileData (:))

ans =

    3.7671
```

Min:

```
>> min(WWB_FileData (:))

ans =

   -0.9326
```

Mean:

```
>> mean(WWB_FileData)

ans =

    3.3250   -0.5884   -0.6887    0.1160
```
If you have questions about this tutorial, please post a support ticket at: http://www.dataq.com/ticket/