

**该内容不适用于您选择的语言**

该内容将以另一种语言显示出来。 您可使用浏览器的翻译功能查看内容。

# Writing TDM and TDMS Files in LabVIEW

Updated Oct 23, 2020

## Environment i

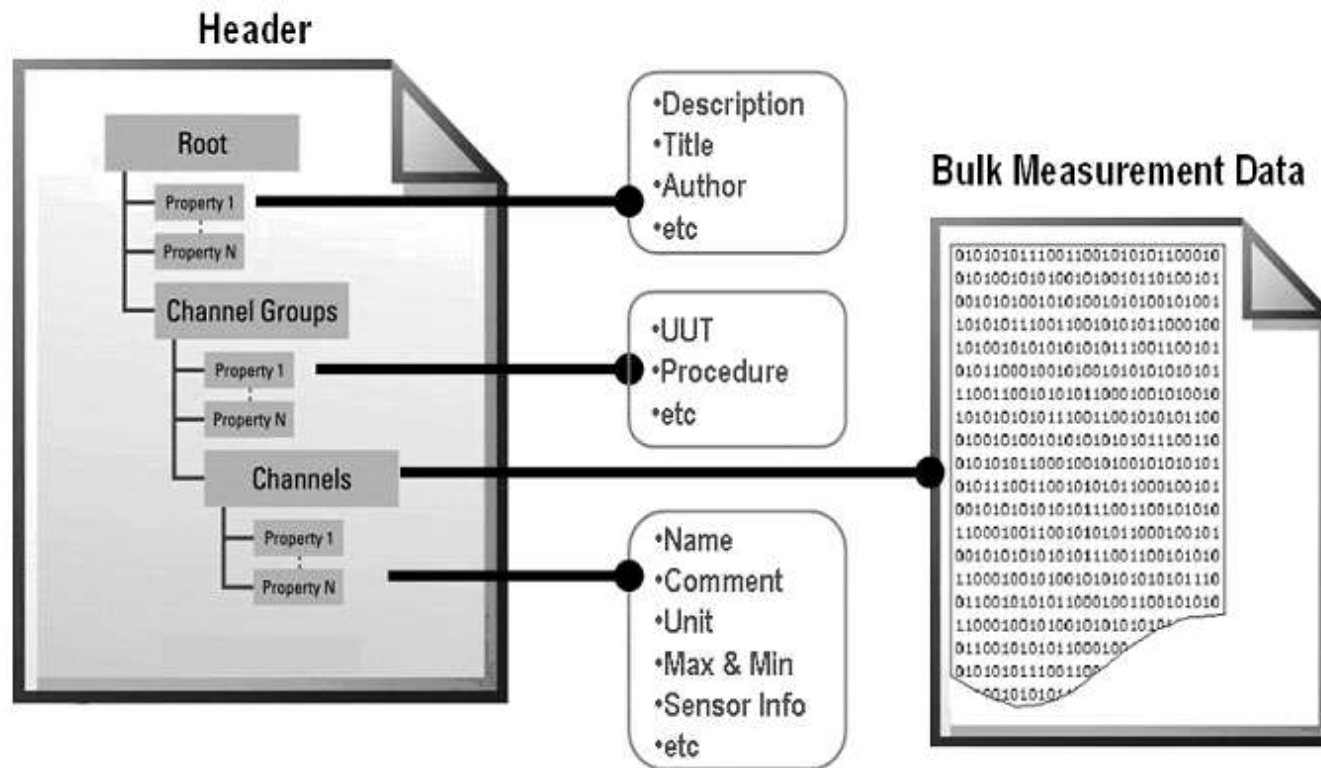
### Software

- LabVIEW

---

To reduce the need to design and maintain your own data file format, NI has created the flexible technical data management (TDM) data model, which is natively accessible through NI LabVIEW, LabWindows™/CVI, Measurement Studio, LabVIEW SignalExpress, and DIAdem and is portable to other common applications such as Excel. The TDM data model offers several unique benefits such as the ability to scale to your specific project requirements and easily attach descriptive information to your measurements while streaming your data to disk.

The TDM data model offers three levels of hierarchy, as shown in Figure 1 – root, group, and channel. Each level accepts an unlimited number of customer-defined attributes, making the files “search ready.”



**Figure 1.** Each TDM and TDMS file written in LabVIEW contains descriptive information on the root, group, and channel levels.

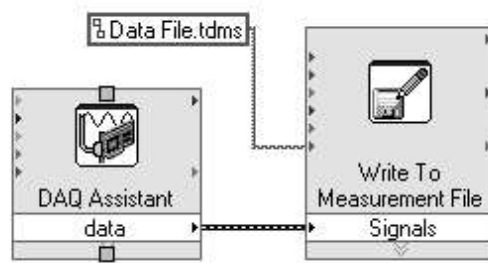
The TDM data model file can contain several groups, and each group can contain several channels. You can insert your own custom properties at each of the three levels.

The TDM data model supports two file formats: TDM and TDMS. The TDM file format specifies that you save descriptive information in a header file with the extension TDM and the bulk measurement, simulation, and analysis results in a bulk binary data file with the extension TDX. Unlike TDM files, which have a strictly required XML-based header file, TDMS files have a binary index file with the extension \*.TDMS\_Index. The TDMS\_Index file provides consolidated information on all the attributes and pointers in the bulk data file, and speeds up access to the data while reading.

In LabVIEW, you can choose from three interfaces to write data to the TDM data model. Some of these interfaces work with both TDM and TDMS file formats. The following sections examine the basics of using each interface.

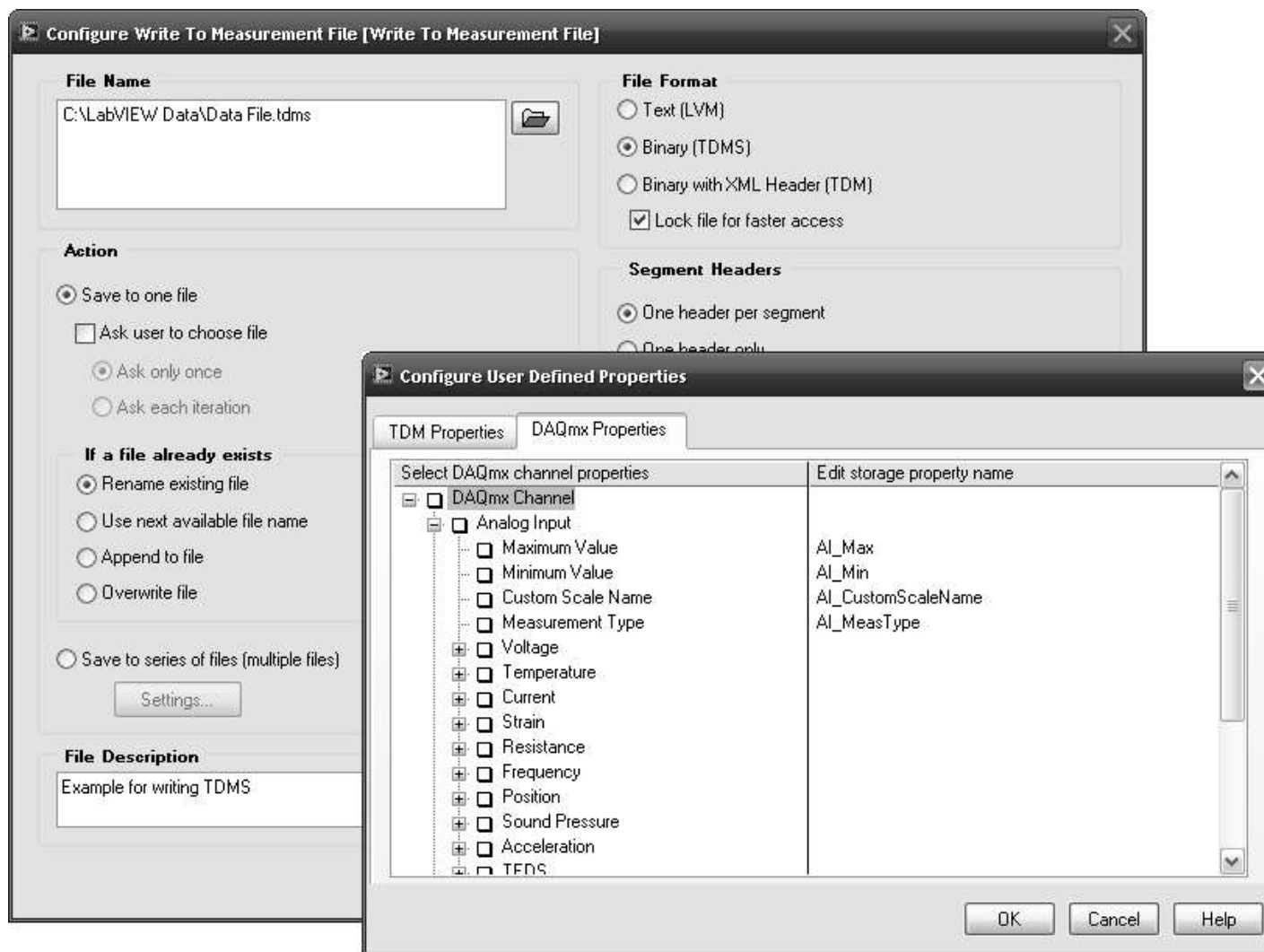
## Write To Measurement File Express VI

The simplest way to write to either a TDM or TDMS file is to use the Write To Measurement File Express VI from the File I/O palette. The only input needed for the VI is the DAQmx Task or the signals you are writing to disk, as shown in Figure 2.



**Figure 2.** You can wire data directly into the Write To Measurement File Express VI.

Within the configuration dialog of the Write To Measurement File Express VI, you can select to write to either a TDM or TDMS file. Also, you can use the configuration dialog to add properties to the file such as description, your own custom properties, or properties defined from the DAQmx Task, as shown in Figure 3.

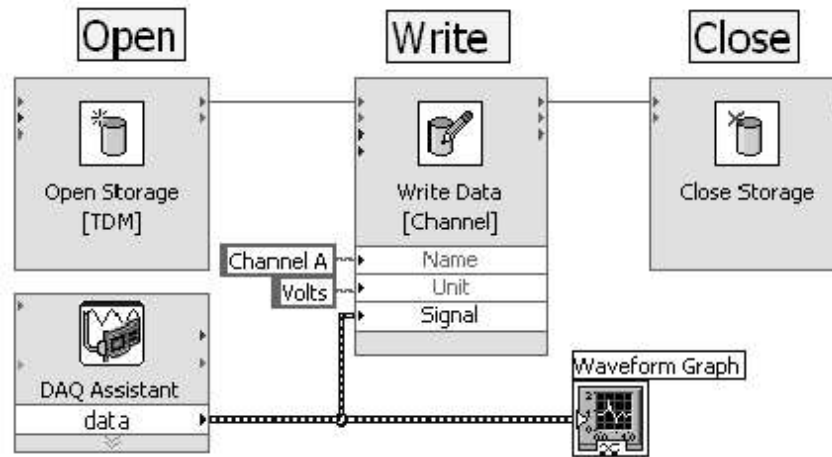


**Figure 3.** After selecting to write to a TDM or TDMS file, you can also configure user-defined properties in the configuration dialog.

The Write To Measurement File Express VI is the simplest way to begin writing data to TDM and TDMS files. However, you have very little flexibility in the structure of the data written. To realize the full benefits of the TDM data model and write well-organized and documented data, you need to use data storage VIs and/or the TDM Streaming API.

## Data Storage VIs

With data storage VIs, you can write data to either a TDM or TDMS file. The simplest form of writing measurement data with the LabVIEW data storage VIs is illustrated in Figure 4, where a data acquisition task generates an array of measurement values to the Write Data VI.

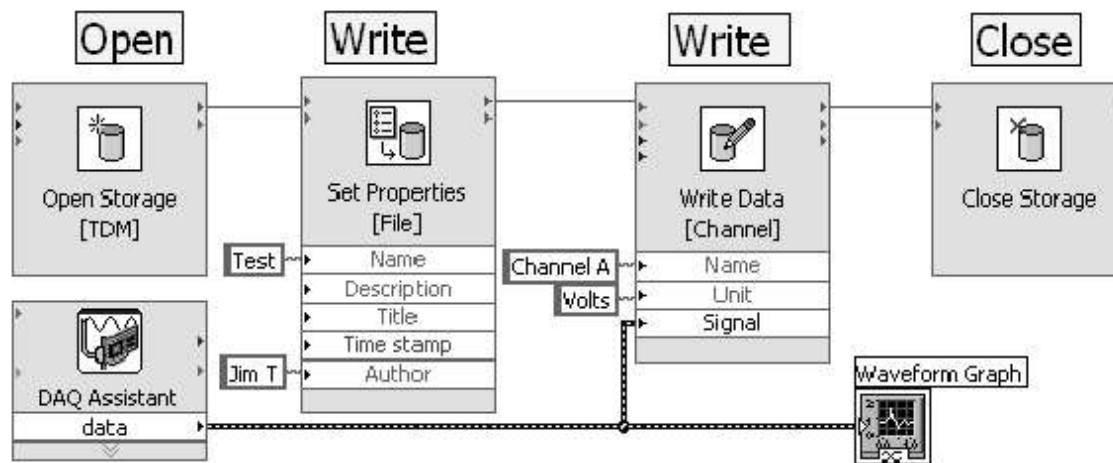


**Figure 4.** Data storage VIs document data at the channel level.

Notice that the Write Data VI has two inputs – name and unit – in addition to the signal input. You choose which additional information you wish to save through a configuration dialog that you access by double-clicking on the Write Data VI, similar to same process for the Write To Measurement File Express VI.

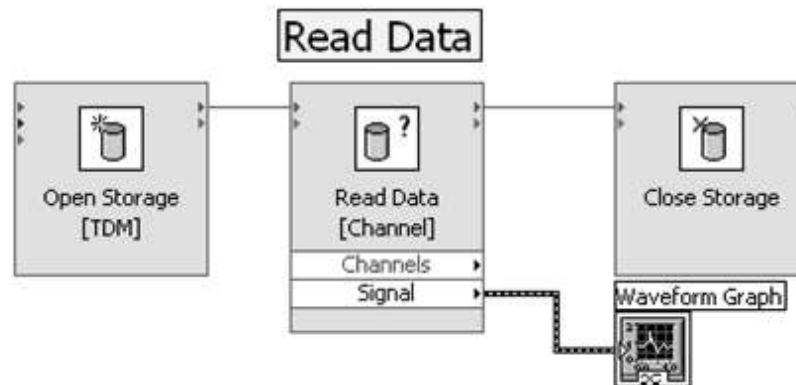
In Figure 4, all of the channels coming from the DAQ Assistant are given the same name and unit; however, the data storage VIs enumerate the channel name to prevent naming collisions. You can override the autoenumeration by passing an array of channel names to the name input or by using an individual Write Data VI for each channel, both of which are acceptable solutions. An additional benefit of the Write Data VI is that it automatically calculates the minimum and maximum values for each measurement array and saves them in the TDM or TDMS file.

Figure 4 outlines how the data storage VIs document data at the channel level. Figure 5 illustrates how you can use the Set Properties VI to save additional descriptive information at the file level.



**Figure 5.** Use the Set Properties VI to save additional descriptive information at the file level.

Figure 6 illustrates the simplest form of reading data using data storage VIs. Here, the Open Storage VI passes a file reference to the Read Data VI and then all data is read from the TDM or TDMS file into memory and displayed on a waveform graph.

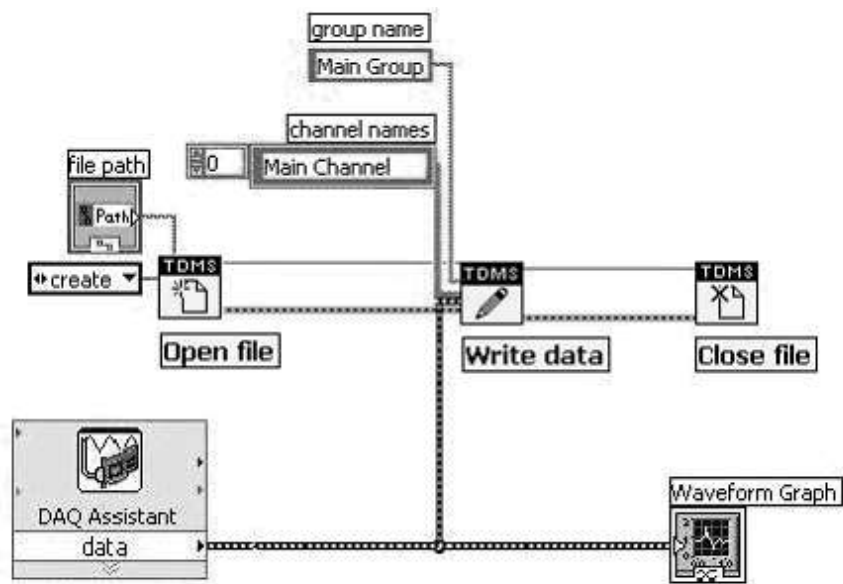


**Figure 6.** The Open Storage VI passes a file reference to the Read Data VI and then all data is read from the TDM or TDMS file into memory and displayed on a waveform graph.

## TDM Streaming API

Although data storage VIs are easy to use and flexible, they do not exhibit the same performance when writing data to disk that you achieve using the TDM Streaming API. The TDM Streaming API was designed to stream data to disk at high speeds and with a small disk footprint, which is why it is also ideal for writing data on a real-time system.

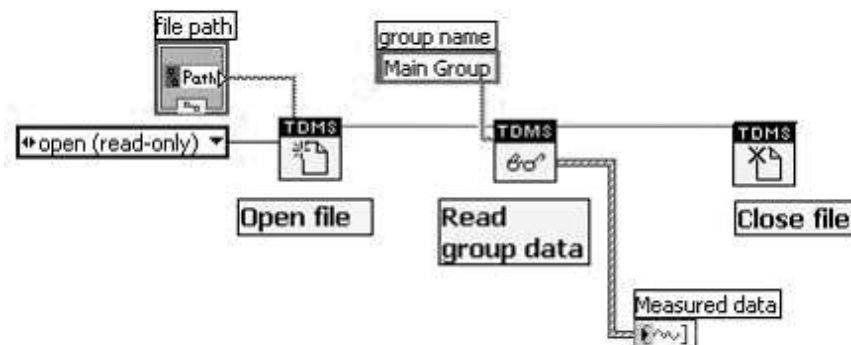
The simplest form of writing measurement data with TDM Streaming VIs is illustrated in Figure 7. In this example, the DAQ Assistant generates an array of measurement values and passes them to the Write Data VI.



**Figure 7.** Write data to a TDMS file at the channel level.

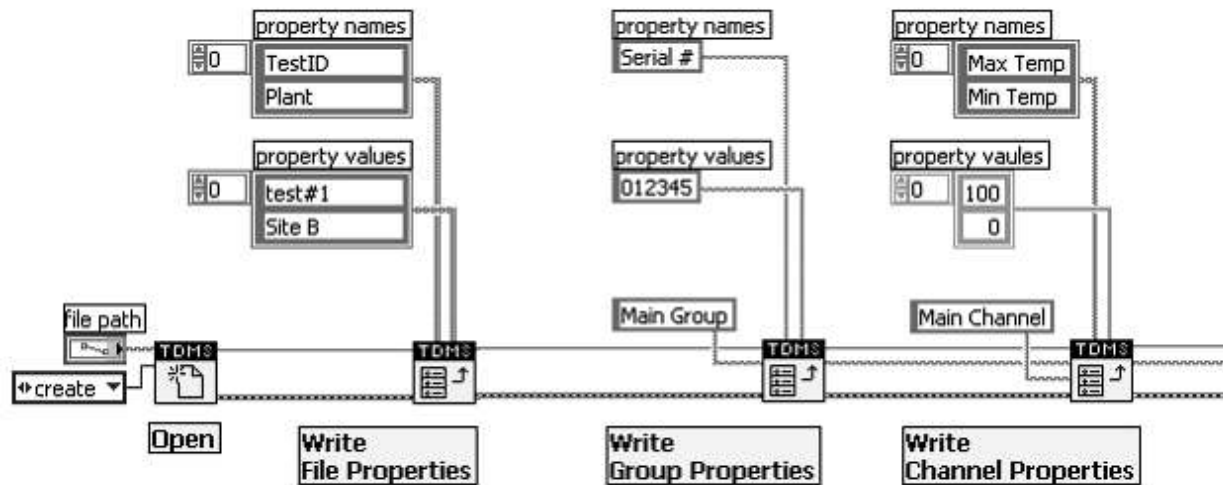
Notice that the Write Data VI has two inputs in addition to the signal input. In Figure 7, all of the channels coming from the DAQ Assistant are given the same group and channel name. If the group and channel name inputs are left unwired, they default to "untitled."

Figure 8 shows the simplest form of reading data using TDM Streaming VIs. Here, the TDMS Open File VI passes a file reference to the TDMS Read File VI and then all of the data in the "Main Group" group is read from the TDMS file into memory and displayed on a waveform graph.



**Figure 8.** Read data back from a TDMS file.

The TDMS data model automatically creates certain properties when some data types are written. However, in many cases, you may want to create a property specific to your particular needs, such as unit under test, serial number, and test temperature. You can achieve this using the TDMS Set Properties VI, with which you can write properties at the file, group, or channel level, as shown in Figure 9.



**Figure 9.** Write custom properties at three different levels before writing data to the file.

The mark LabWindows is used under a license from Microsoft Corporation. Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

## Related Links

- [Introduction to LabVIEW TDM Streaming VIs](#)
- [Explore More NI Data Management Platform Tools](#)

## Other Support Options

### Ask the NI Community

Collaborate with other users in our discussion forums

- [Search the NI Community for a solution](#)

### Request Support from an Engineer

A valid service agreement may be required, and support options vary by country.



- [Open a service request](#)
- [Purchase or renew support services](#)