

Document Type: Tutorial

NI Supported: Yes

Publish Date: Nov 5, 2009

Introduction to LabVIEW TDM Streaming VIs

Table of Contents

1. Introduction
2. The TDMS File Format of the TDM Data Model
3. LabVIEW TDM Streaming API for Writing and Reading TDM Files
4. TDMS Writing Data Basics
5. TDMS Reading Data Basics
6. TDMS Write Custom Properties
7. TDMS Read Custom Properties
8. TDMS File Viewer
9. TDMS File Defragment and TDMS Flush
10. Additional TDMS Information

Introduction

To reduce the need to design and maintain your own data file format, National Instruments has created a flexible data model called TDM, which is natively accessible through NI LabVIEW, LabWindows/CVI, and DIAdem, and is portable to other applications such as Excel. A [TDM DLL](#) is also available for accessing the TDM data model with third-party software applications. 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 supports two file formats, TDM and TDMS. The TDM Streaming API, the subject of this paper, writes TDMS files and is optimized for streaming well documented data to disk. The other file format of the TDM data model, written by the [Data Storage VIs](#) and the Write to Measurement File express VI, offers the same flexibility with regard to documenting data, but is not optimized for streaming data to disk. The new TDM Streaming VIs offer the widest coverage of different use cases, and therefore are the recommended way to interface with the TDM data model. However, the [Data Storage VIs](#) do offer some unique capabilities, including the ability to read and write TDMS files as well as querying data from a file. The [TDM white paper](#) provides a comprehensive overview of the NI TDM Platform, including both APIs.

Read this paper to learn more about the TDM data model and see how LabVIEW TDM Streaming VIs reduce your programming time while increasing the value of the data you are collecting.

The TDMS File Format of the TDM Data Model

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”.

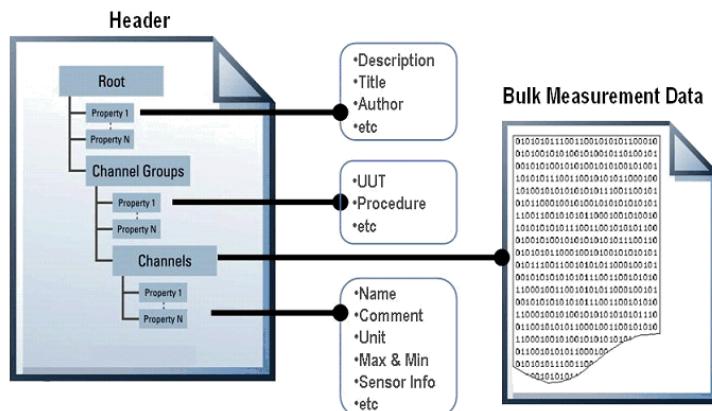


Figure 1. Each TDMS file written in LabVIEW contains descriptive information on the root, group, and channel levels.

Unlike the 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. The TDMS_Index file is automatically regenerated if lost.

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 descriptive information located in the TDMS file, a key benefit of this format, provides an easy way to document the data without having to design your own header structure. As your documentation requirements increase, you do not have to redesign your application; you simply extend the TDMS data model to meet your specific needs.

LabVIEW TDM Streaming API for Writing and Reading TDM Files

Like other file I/O functions in LabVIEW, you have to open, write to, and close the file with the TDM Streaming API. You can also use the native TDMS VIs from the palette to save the properties you choose rather than worrying about file formatting, data types, and other tasks normally associated with writing additional information into a measurement file. The following sections step you through the basics of writing and reading, creating properties, and sharing your TDMS files.

TDMS Writing Data Basics

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

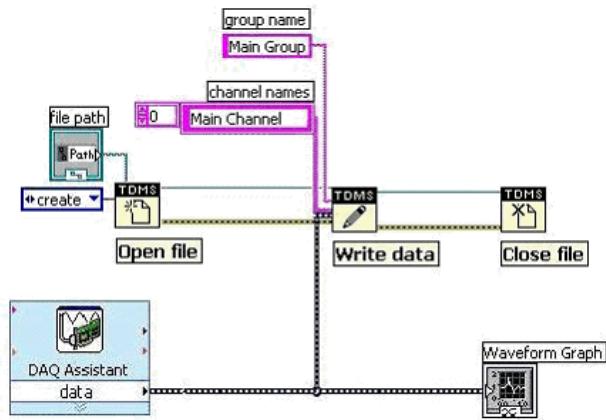


Figure 2. Write data to a TDMS file at the channel level.

Notice that the Write Data VI has two inputs in addition to the Signal input. In the example above, all the channels coming from DAQ Assistant are given the same group and channel name. If the Group and Channel Name inputs are left unwired, they will default to "Untitled".

TDMS Reading Data Basics

Figure 3 illustrates 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 data in the "Main Group" group is read from the TDMS file into memory and displayed on a waveform graph.

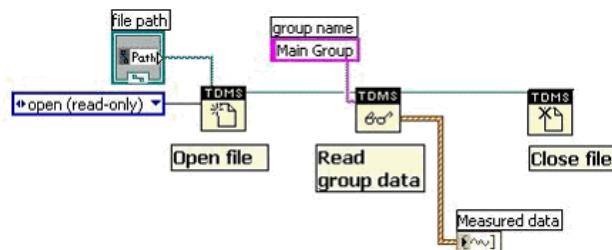


Figure 3. Read data back from a TDMS file.

TDMS Write Custom Properties

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 UUT, serial number, and test temperature. This task can be accomplished using the TDMS Set Properties VI with which you can write properties at the file, group, or channel level, as shown in Figure 4.

The file level of a property determines which input terminals need to be wired. For example, to write a group property, only the Group Name terminal needs to be wired. To write a channel property, both the Group Name and Channel Name terminals should be wired. But, to write a file property, the Group Name and Channel Names terminals should be left unwired. These cases are illustrated in Figure 5.

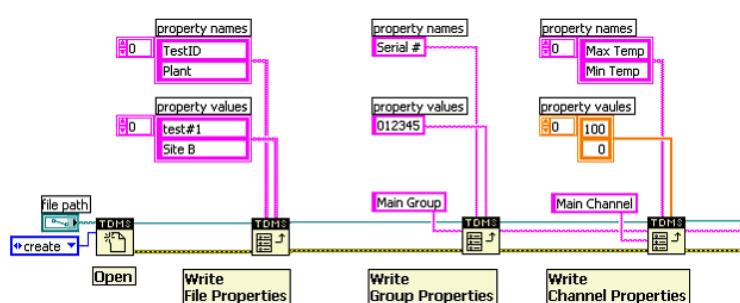


Figure 4: Write custom properties at three different levels before writing data to the file.

With the TDMS Set Properties VI, you can specify an individual property by itself or specify many properties by using arrays. Figure 4 shows that we are specifying two properties at the file level (TestID and Plant). This could be expanded to specify many more properties by just increasing the size of the array. Arrays are not necessarily needed if only a single property is written, such as a serial number.

Property values can also be different data types. In Figure 4, string properties were written at the file and group level. But at the channel level, we write two numeric properties to specify the minimum and maximum temperature.

Figure 5 illustrates how you can use the TDMS Set Properties VI to save additional descriptive information at the file level. In this example the date and the user of the test are built into an array and then written as custom properties.

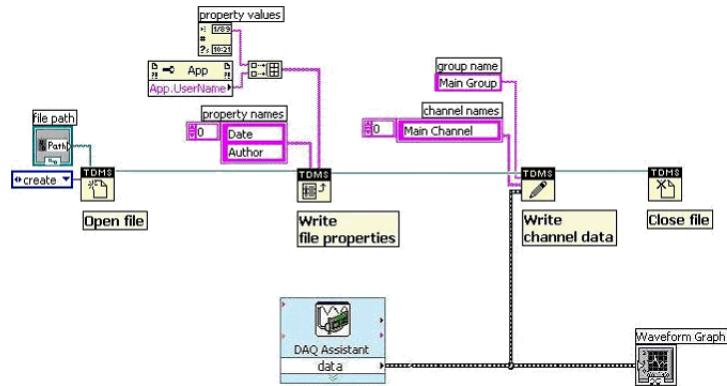


Figure 5. Use the Set Properties VI to save additional descriptive information at the file level.

TDMS Read Custom Properties

Once a TDMS File has been written, the properties can be read back into LabVIEW using the TDMS Get Properties VI. Properties will be returned for only the level specified by wiring the Group Name and Channel Name terminals. For example, Figure 6 shows how to get properties for the file level, because neither the Group Name nor the Channel Name is wired. But to retrieve the properties for a particular channel, both the name of the channel group and the name of the channel need to be connected to the Group Name and Channel Name terminals. This process is very similar to writing the properties, and it is shown in Figure 6.

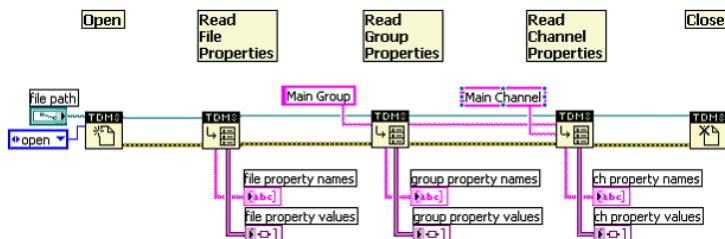


Figure 6. Read TDMS Properties from three levels in the data storage model.

In this configuration, the property values are returned as a variant array because the data could be a string, double, Boolean, etc. The data can be displayed on the front panel as a variant or it can be converted in LabVIEW to the appropriate data type. If a property name and its data type are known, they can be wired as inputs to the TDMS Get Properties VI and read directly with the correct data type.

TDMS File Viewer

The TDMS File Viewer is a powerful tool to use when developing a TDMS application. With it you can automatically see everything that has been written to a TDMS file by making a single VI call. It is very flexible and can read very complex TDMS files.

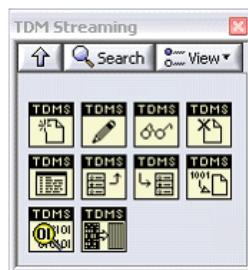


Figure 7. Launch the TDMS File Viewer by placing it after the TDMS file has been closed.

The file viewer has been included with the other TDM Streaming VIs so it can be easily placed in a program. It is located in the functions palette with the other TDM Streaming VIs, as shown in Figure 7. It can be used by placing it after the file is closed, as shown in Figure 8. It will automatically launch another window in which you can view the data and properties inside the TDMS file.

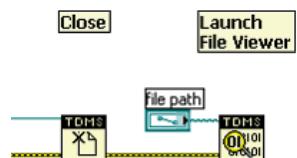


Figure 8. Launch the TDMS File Viewer by placing it after the TDMS file has been closed.

Figure 9 shows an example of what the File Viewer looks like. The left side of the File Viewer has a tree structure so you can access the different levels of data in the file such as groups and channels. The right side of the window displays the properties and data inside of the level that is selected from the tree structure on the left.

There are four tabs that appear in the right side of the window. The Property tab displays all properties associated with a certain data storage level. The Values tab displays all written data in a table format. The Analog Values and Digital Values tabs display the data in a graphical format.

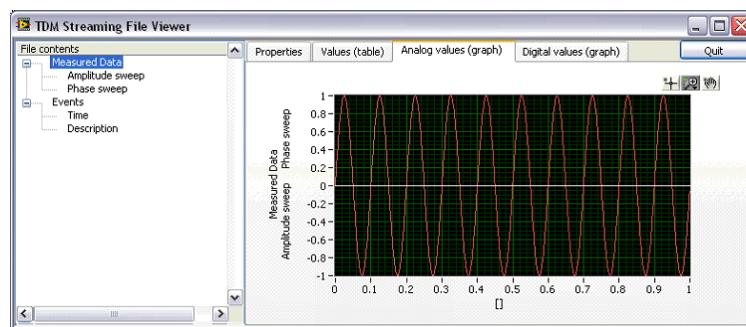


Figure 9. The TDMS File Viewer provides easy access to your TDMS data file.

TDMS File Defragment and TDMS Flush

Included with the TDM Streaming VIs are the defragment and flush features for optimizing read and write operations.

Each call of the TDMS File Write VI records a block of data to file, which may contain several channels. This prevents the data for a single channel from being written contiguously across multiple writes. The TDMS File Defragment VI reorganizes the file by channel to optimize read operations. In this way, the user can take the performance penalty only once by defragmenting the TDMS file "offline" rather than with every read. For example, the user is able to acquire and store data very rapidly and then defragment the file before performing any read operations.

The TDMS Flush VI forces all cached data to disk, a useful feature to ensure that no data will be lost in the event of a system crash. Using the TDMS Flush VI will cost some system performance.

Additional TDMS Information

The LabVIEW TDM Streaming VIs are designed to write data quickly and efficiently to the disk. They were also designed to work on real-time systems so that data can be quickly stored on a deterministic system. Therefore, with the TDM Streaming VIs, you can keep track of your data in a structured format on a real-time system instead of writing data in a binary format with no predefined framework.

TDMS files are native in National Instruments LabVIEW, LabWindows/CVI, and DIAdem software. DIAdem software uses data mining technology to easily find relationships and correlations in your data. An add-in is also available to load TDMS files into Microsoft Excel. Visit ni.com/tdm to download the Excel TDMS add-in.

Conclusion

LabVIEW offers a variety of file I/O options for reading and writing data. The TDM Streaming VIs provide several benefits that go beyond traditional file operations in LabVIEW. These benefits include:

1. Stream data to disk quickly in a structured format
2. Easily save descriptive information with test data, which makes it search-ready
3. Categorize data under groups to add structure (for example, Group 1 = temperature sensors, Group 2 = pressure sensors)
4. Easily use the API to abstract the low-level file I/O normally associated with saving data
5. Include properties to help describe data when analyzed later
6. Use in real-time systems

Legal

This tutorial (this "tutorial") was developed by National Instruments ("NI"). Although technical support of this tutorial may be made available by National Instruments, the content in this tutorial may not be completely tested and verified, and NI does not guarantee its quality in any way or that NI will continue to support this content with each new revision of related products and drivers. THIS TUTORIAL IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND AND SUBJECT TO CERTAIN RESTRICTIONS AS MORE SPECIFICALLY SET FORTH IN NI.COM'S TERMS OF USE (<http://ni.com/legal/termsofuse/unitedstates/us/>).