How to Choose the Right Data Storage Format for Your Measurement System

Overview

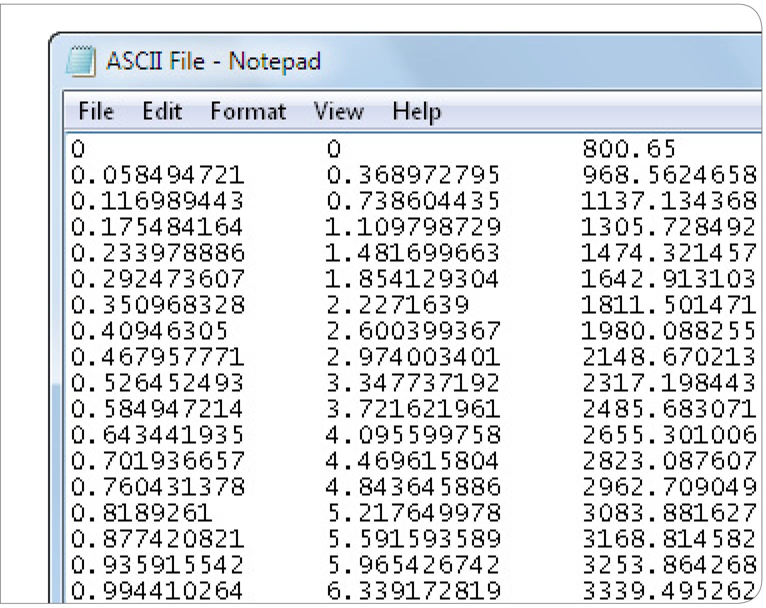
For many new measurement systems, choosing the right data storage approach is an afterthought. Engineers often end up selecting the storage strategy that most easily meets the needs of the application in its current state without considering future requirements. Yet storage format choices can have a large impact on the overall efficiency of the acquisition system as well as the efficiency of postprocessing the raw data over time.

Today's complex products require data acquisition throughout the complete design and development process. Companies make a significant investment in the data they collect. Increasing microprocessor speed and storage capacity together with decreasing costs for hardware and software have resulted in an explosion of data stored in files and databases. While technology is enabling faster and richer data retention, managing and making good use of this data remains the real challenge. This white paper covers five things to consider when choosing your data storage format.

- Can my data storage format handle my data volume and data streaming speeds?
- Do I need to exchange data with my colleagues?
- Can my analysis or visualization software use my storage format?
- Is my storage format flexible enough for my future needs?
- How do I retrieve my stored data at a future date?

Can my data storage format handle my data volume and data streaming speeds?

The first step to achieving a cohesive data management solution is ensuring that data is stored in the most efficient, organized, and scalable fashion. All too often data is stored without descriptive information, in inconsistent formats, and scattered about on arrays of computers, which creates a graveyard of information that makes it extremely difficult to locate a particular data set and derive decisions from it.



```
0               0               800.65
0.058494721     0.368972795     968.5624658
0.116989443     0.738604435     1137.134368
0.175484164     1.109798729     1305.728492
0.233978886     1.481699663     1474.321457
0.292473607     1.854129304     1642.913103
0.350968328     2.2271639       1811.501471
0.40946305      2.600399367     1980.088255
0.467957771     2.974003401     2148.670213
0.526452493     3.347737192     2317.198443
0.584947214     3.721621961     2485.683071
0.643441935     4.095599758     2655.301006
0.701936657     4.469615804     2823.087607
0.760431378     4.843645886     2962.709049
0.8189261       5.217649978     3083.881627
0.877420821     5.591593589     3168.814582
0.935915542     5.965426742     3253.864268
0.994410264     6.339172819     3339.495262
```

Figure 1.
ASCII files are easy to exchange but can be too slow and large for many applications.

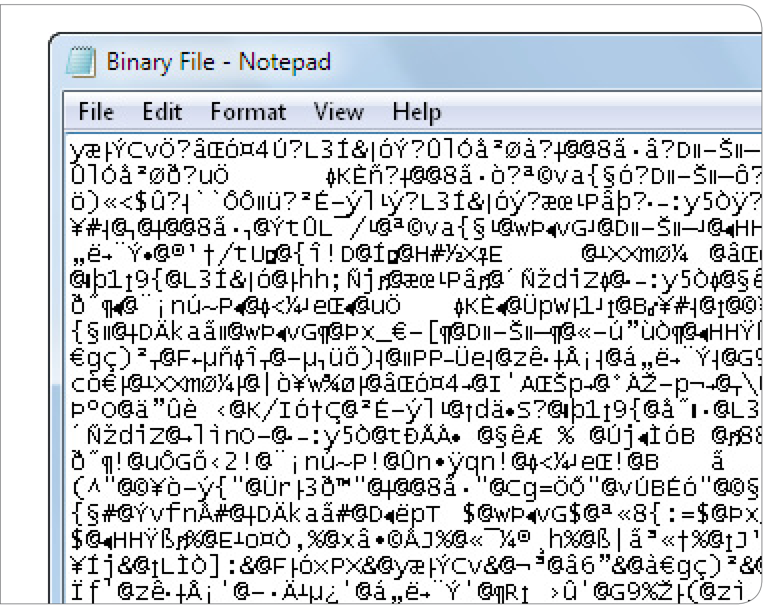
Depending on the application, you may prioritize certain characteristics over others. Common storage formats such as ASCII, binary, and XML have strengths and weaknesses in different areas.

ASCII Files

Many engineers prefer to store data using ASCII files because of the format's easy exchangeability and human readability. ASCII files, however, have several drawbacks, including a large disk footprint, which can be an issue when storage space is limited (for example, storing data on a distributed system). Reading and writing data from an ASCII file can be significantly slower compared to other formats. In many cases, the write speed of an ASCII file cannot keep up with the speeds of acquisition systems, which can lead to data loss.

Binary Files

Another typical storage approach that is somewhat on the opposite end of the spectrum from ASCII is binary files. In contrast to ASCII files, binary files feature a significantly smaller disk footprint and can be streamed to disk at extremely high speeds, making them ideal for high-channel-count and real-time applications. A drawback to using binary is its unreadable format that complicates exchangeability between users.



```
yæ╞ÝCvÖ?âŒó¤4Ú?L3Í&|óÝ?Ù]óå²øà?╪@@8ã·â?D‖–Š‖–
Ù]óå²øð?uö        ╪KÈñ?╪@@8ã·ò?ª©va{§ó?D‖–Š‖–ô?
ö)«<$û?┤``ÔÔ‖ü?²É–ý]└ý?L3Í&|óý?æœ└Pâþ?·–:y5Òÿ?
¥#┤@┐@╪@@8ã·┐@ÝtÙL¯/└@ª©va{§└@wP◄vG┘@D‖–Š‖–┘@◄HH
„ë→¨Ý•@®'†/tU◘@{î!D@Í◘@H#½×‡E       @┴××m0¼  @âŒ
@▌þ1↕9{@L3Í&|ó@╞hh;Ñj╓@æœ└Pâ╓@´ÑždiZ╪@→–:y5Ò╪@§ê
ð˝¶◄@¨¡nú~P◄@╪<¾┘eŒ◄@uÖ      ╪KÈ◄@ÜpW╞1┘↕@B▫¥#┤@↕@©
{§‖@╪DÄkaã‖@wP◄vG¶@Px_€–[¶@D‖–Š‖–¶@«–ú”ùÒ¶@◄HHŸ
€gç)²┬@F→µñ╪î┬@–µ┐üõ)┤@‖PP–Üe┤@zê·┼Å¡┤@á„ë→¨Ý┤@G9
cö€▐@┴××m0¼▐@|ò¥w¾0▐@âŒó¤4→@I'AŒŠp→@˚ÅŽ–p¬→@┬\
Þºo@ä”Ûè <@K/IótÇ@²É–ý]└@↕dä•S?@▌þ1↕9{@å˝▮·@L3
´ÑždiZ@→]inO–@→–:y5Ò@tÐÄÅ• @§êÆ % @Új◄ÍóB @╓8
ð˝¶!@uÔGő‹2!@¨¡nú~P!@Ûn•ÿqn!@╪<¾┘eŒ!@B    ã
(^"@©¥ò–ý{"@Ür╞3ð™"@╪@@8ã·"@Cg=ÖŐ"@vÚBÉó"@©§
{§#@ÝvfnÅ#@╪DÄkaã#@D◄ëpT  $@wP◄vG$@ª«8{:=$@Px
$@◄HHŸß╓%@E┴o¤Ò,%@xâ•@ÅJ%@«¯¾®¸h%@ß|ã³«†%@↕J'
¥Íj&@↕LÌÒ]:&@F╞óxPX&@yæ╞ÝCv&@¬³@â6”&@à€gç)²&
Ïf'@zê·┼Å¡'@–·Ä┴µ¿'@á„ë→¨Ý'@¶R↕ ›Û'@G9%Ž╞(@zi
```

Figure 2.
Binary files are beneficial in high-speed, limited-space applications but can cause exchangeability issues.

XML Files

Over the last several years, the XML format has been gaining in popularity due to its ability to store complex data structures. With XML files, you can store data and formatting along with the raw measurement values. Using the flexibility of the XML format, you can store additional information with the data in a structured manner. XML is also relatively human-readable and exchangeable. Similar to ASCII, XML files can be opened in many common text editors as well as

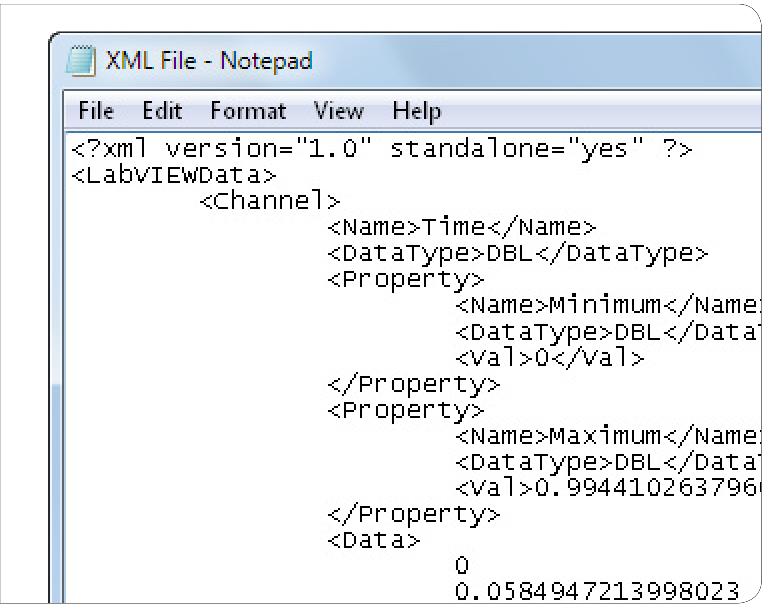
XML-capable Internet browsers, such as Microsoft Internet Explorer.

Database Files

Database files are composed of a series of tables, built using columns and rows, and information may or may not be linked between tables. Though databases are advantageous because they are searchable, database files are impractical for time-based measurement applications given the amount of data acquired and the need to either purchase or build a formal database solution from scratch.

TDMS Files

TDMS is a binary-based file format, so it has a small disk footprint and can stream data to disk at high speeds. At the same time, TDMS files contain a header component that stores descriptive information, or attributes, with the data. Some attributes such as file name, date, and file path are stored automatically; however, you can easily add your own custom attributes as well. Another advantage of the TDMS file format is the built-in three-level hierarchy: file, group, and channel levels. A TDMS file can contain an unlimited number of groups, and each group can contain an unlimited number of channels. You can add attributes at each of these levels describing and documenting your test data for better understanding. This hierarchy creates an inherent organization of your test data.



```
<?xml version="1.0" standalone="yes" ?>
<LabVIEWData>
        <Channel>
                <Name>Time</Name>
                <DataType>DBL</DataType>
                <Property>
                        <Name>Minimum</Name
                        <DataType>DBL</Data
                        <Val>0</Val>
                </Property>
                <Property>
                        <Name>Maximum</Name
                        <DataType>DBL</Data
                        <Val>0.994410263796
                </Property>
                <Data>
                        0
                        0.0584947213998023
```

Figure 3.
XML files can help you define complex structures but are significantly larger and slower than other formats.

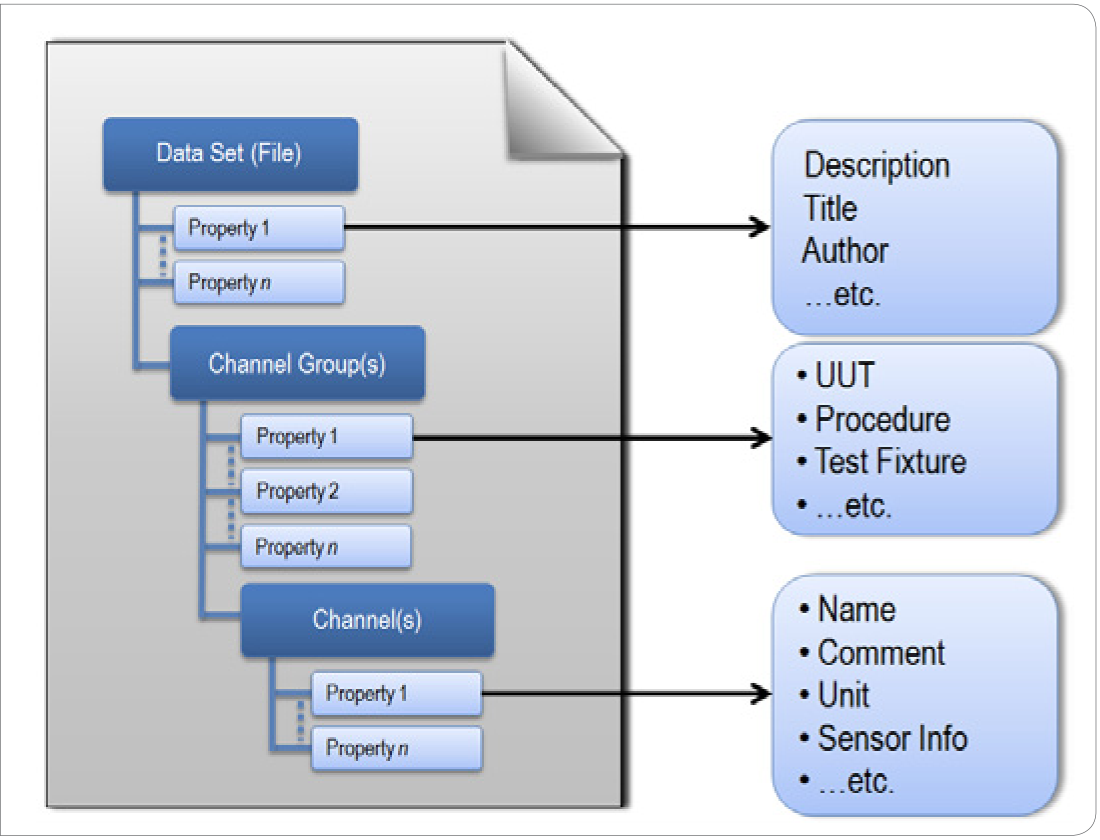


Figure 4.
The hierarchical TDMS file format is designed to meet the needs of engineers collecting measurement data.

Do I need to exchange data with my colleagues?

Often simulation and test systems grow over several years, usually independently of each other and with equipment from different suppliers. As a result, data is stored in different file formats, with little or no descriptive information and in different locations. All of these factors present roadblocks that impede the optimal exchange of information and make it extremely difficult to locate a particular data set and derive decisions from it. As a result, many companies are seeing a loss in efficiency and increasing development costs as the amount of data they store increases.

ASCII Files

Many engineers prefer to store data using ASCII files because of the format's easy exchangeability and human readability. ASCII files make it simple to quickly open files written from acquisitions and view data immediately as well as to easily share the data with colleagues because the files can be opened in common software applications found on most computers today such as Notepad, WordPad, and Microsoft Excel.

Binary Files

The shortcoming of saving data to a custom binary file format is that it is not human-readable and therefore difficult to exchange between users. Binary files cannot be immediately opened by common software; they have to be interpreted by a specialized application or program. Different applications may interpret binary data in different ways, which causes confusion. To share the files with colleagues, you must provide them with an application that interprets your specific binary file correctly. Also, if you make changes to how the data is written in the acquisition application, these changes must be made in the application that is reading data. This can potentially cause long-term application versioning issues and headaches that can ultimately result in lost data.

XML Files

In its raw form, XML includes tags within the file that describe the structures. These tags also appear when XML files are opened in these applications, which somewhat limits the readability because you must be able to understand these tags. This may be a challenge for colleagues less involved in the file architecture development. Also, a downside to being able to store these complex structures is that they may require considerable planning when you design the layout, or schema, of the XML structures.

Database Files

Database files can be shared among team members and usually opened via common applications. With measurement data, they have limitations similar to ASCII and XML files: large footprint, risk of lost data, and limited streaming to disk.

TDMS Files

Although TDMS files are binary, you can open them in many common applications, such as Microsoft Excel and OpenOffice Calc, for sharing with colleagues. TDMS files give you the benefits of easy exchangeability and attribute inclusion without sacrificing speed and size.

Can my analysis or visualization software use my storage format?

When choosing a file format, you need to know which formats your analysis or visualization software can accept.

ASCII Files

ASCII files can be opened by nearly every data analysis and visualization package on the market, but, as is typical of the format, they are limited in their disk read speeds. Processing and visualizing ASCII files is drastically slower than their binary equivalent.

Binary Files

Custom binary files need to be changed and updated over time, which means that analysis and visualization software needs to accommodate these version changes. When application software is updated to interpret the binary file, analysis and visualization speeds are radically improved using the binary format since the software doesn't need to handle ASCII overhead.

XML Files

XML is built on a tree model and, therefore, mapping the trees to the analysis or visualization API is required. This mapping can be difficult and complex, potentially causing delays and introducing errors into your project. In addition, errors you encounter while processing an XML file can disrupt your postprocessing and delay your project.

Database Files

Database files, which are constructed in a structured fashion similar to XML files, face the same challenges in terms of mapping the content and being increasingly complicated as your data set gets larger and more diverse. Application software must be able to interface with the database to read and potentially convert the data sets before analysis or visualization.

TDMS Files

Since TDMS is a binary file format, analyzing or visualizing data from a TDMS file is extremely efficient. TDMS is an open standard that can use plug-ins or a free DLL to load TDMS data into a variety of different software packages.

Is my storage format flexible enough for my future needs?

You can choose from a variety of format options for measurement data storage. Unfortunately, careful consideration of data storage options is not typically at the forefront of application planning. The file format choice is often overlooked in favor of higher-visibility decisions such as hardware system design or software architecture. Data storage decisions are sometimes made arbitrarily or on an as-needed, per-application basis without second thought for reusability and scalability. This leads to complex and costly software rearchitecture. You should carefully consider the scalability of your data storage format choice to ensure reuse for future requirement changes.

How do I retrieve my stored data at a future date?

More than ever in today's fiercely competitive business environment, companies need to rapidly turn test and simulation data into usable information to efficiently drive product development and reduce time to market. In many acquisition systems, the application is designed to quickly collect the raw data and store it to disk, possibly with some inline processing for decision making. However, to truly move from the raw data to usable results, you need tools to quickly find trends and outliers in the data, analyze these trends, and report them in a way that can be shared with others.

Only database and TDMS files are searchable without having to open each file and manually search for the data you need. Depending on how a database file is structured, the search process can be slow. In addition, the database files you want to search need to be in the same location and structured exactly the same. With diverse teams across the world, implementing a strict database file structure can be virtually impossible as every engineer, department, and company already has its own data management policy that is likely different from one another.

In contrast, TDMS files are searchable and do not need to be formatted and stored in precisely the same location. In fact, the more custom properties you use to document your measurement data, the more easily you can locate it at a later date. The descriptive information located in the file, a key benefit of TDMS, provides an easy way to document the data without you having to design your own header structure. As your documentation requirements increase, you do not have to redesign your application; you simply extend the model to meet your specific needs.

Summary

	ASCII	Binary	XML	Database	TDMS
Exchangeable	✓		✓		✓
Small Disk Footprint		✓			✓
Searchable				✓	✓
Inherent Attributes			✓		✓
High-Speed Streaming		✓			✓
NI Platform Supported	✓	✓	✓	✓*	✓

*May require a toolkit or add-on module.

Table 1.
The TDMS file format combines the benefits of several data storage options in one file format.

Several roadblocks can impede the optimal exchange of technical information. The most notorious is the improper storage of information at the time of test or simulation. You should always make sure that your data is stored with descriptive information, in consistent formats, and easily locatable. Table 1 summarizes the pros and cons of some of the most commonly chosen storage options for measurement data.

▶ **Read about data management and mining with NI DIAdem**