Presented to the 2008 Georgia Tech Fault and Disturbance Analysis Conference
Atlanta, Georgia (May 20th, 2008)

Summary Paper of IEEE Std. C37.232-2007
Recommended Practice for Naming Time Sequenced Data Files

Sponsored by the Power System Relaying Committee (PSRC) of the Power Engineering Society (PES) of the Institute of Electrical and Electronics Engineers (IEEE)

Members (H8-WG): Mark Adamiak, John Chadwick, Rick Cornelison, Ratan Das, Anthony Eshpeter, Tony Giuliante, Erich Gunther, Jim Hackett, Jim Ingleson, Robert Johnson, Mladen Kezunovic, Amir Makki, Ken Martin, Pierre Martin, Harish Mehta, Krish Narendra, Robert Orndorff, Jeff Pond, Maria Rothweiler, George Semati, Larry Smith, John Sperr, Bill Strang, Mark Taylor, Don Wardlow.

Definition: A time sequence data (TSD) file is any type of electronic data file where each data item in the file corresponds to an instant of time that is identified by an explicit or implicit time tag, such as files produced by digital relays, fault recorders, power swing recorders, power quality monitors, and so on.

Abstract: This paper is a summary of the IEEE Std. C37.232-2007 which recommends a procedure for naming TSD files. The summary includes a description of the reasons and history behind the development of the standard. The advantages of using the standard are highlighted and the limitations and applications are identified. Issues of compatibility across operating systems and adaptability to other types of files are also discussed. The required and optional portions of the naming practice are described in detail and a number of examples are provided.

Keywords: TSD, filename, file name, file extension, CSV, required fields, user fields, start date, start time, fault date, fault time, time code, station identifier, device identifier, company name.

INTRODUCTION

Microprocessor based measurement and protection devices produce large quantities of TSD files. Additional TSD files are also produced while processing raw data and while performing maintenance and testing operations. Each TSD file is identified by a filename that is either assigned manually by the user or is assigned automatically by the originating device. Clearly, unless a common naming practice is in use then there will be many problems handling such large quantities of TSD files (such as sorting, saving, reporting, processing, managing, exchanging, archiving and so forth).

For example, after the August 14th blackout of 2003, the North American Electric Reliability Council (NERC) began an investigation into the root causes behind the blackout with the aim of making recommendations to help avoid such events in the future. To that

extent, NERC collected thousands of digital fault data and event records (TSD files) from many of the involved utilities. The collected data files did not have a common naming convention and because of that it became difficult to discern which files came from which utilities and which ones were captured by which devices. The lack of a common naming practice seriously hindered the investigation process. Subsequently, and in their initial report on the blackout, NERC stressed the need for having a common naming practice and listed it as one of their top ten recommendations.

Fortunately, and since 1999, the H8 working group of the PSRC communications subcommittee was already developing a recommended naming practice and their initial report was published in 2001. NERC adopted the H8 report and renamed their files accordingly. The report helped solve many of the problems associated with the handling of such large quantities of data files and was duly noted by NERC and by NPCC (the Northeast Power Coordinating Council) as well. Finally, in August of 2007, the naming practice was approved by the IEEE Standards Association and is now published under IEEE Std C37.232-2007.

The naming practice defines a human readable filename that includes, among other features, key portions of the information contained in the file contents including, but not limited to, the name of the circuit, the name of the substation, the name of the recording device, and the date and time of event occurrence. The filename information is specified in comma-delimited format where commas are used to separate the information into multiple fields. Thus, spreadsheet like tables can be made from directory listings of filenames. These tables provide users with an easy way to perform sort and query operations based on any one of the fields in the filename, in effect, providing the same look and feel as any other real life database. The naming practice is quickly gaining popularity and is now being used industry wide by a nontrivial number of utilities, system operators, manufacturers, and third party providers.

In what follows, a brief background on filenames is provided along with a description of a number of old proprietary naming practices that are still in circulation today. In addition, the standard naming practice is defined and a number of useful examples are provided. In closing, the limitations and applications of the new standard are discussed in relation to the trends and objectives of this work moving forward.

BACKGROUND

A “filename” is composed of two parts: the first part is the “name” and the second part is the “extension”. The two parts are separated using the dot “.” symbol. The extension is normally used to specify the file type. Typically speaking, there are two types of files: the first type is binary, such as, program files (.EXE) and dynamic link libraries (.DLL), and the second type is ASCII, such as, initialization files (.INI) and text files (.TXT). Filenames are listed in hidden files called allocation tables (File Allocation Table "FAT" for DOS or NT File System "NTFS" for Windows NT). Each entry in these allocation tables corresponds to a single file and has a number of fields including, but not limited to, the name and

extension, the attributes ("A" for archive or "H" for hidden), and the memory addresses to where the file's contents are stored. Historically, the DOS style filenames have been the most popular and are the most restrictive. However, Windows significantly relaxed these restrictions. A Windows style filename can be up to 253 characters long (11 for DOS) and each filename character can be any one of 245 ASCII codes (52 for DOS).

Meaningful filenames are essential for understanding content. Without meaningful filenames, the user is forced to open each file to see its contents. Reading contents, as such, requires considerable disk access time and especially so for large files. However, reading just the filenames is much faster because they are stored separately from their contents in tables that can be quickly loaded with minimal disk access time. Therefore, meaningful filenames provide users and developers with the ability to use and write programs that can quickly manage and process large numbers of files. The hardware benefits, too, from the reduced number of disk access operations. The alternative to meaningful filenames is to build and maintain specialized databases which are very costly to create and do require substantial programming and maintenance support.

PROPRIETARY NAMING PRACTICES

A number of old, proprietary naming practices are still in circulation today because the majority of the recorders in the field were actually built in the late 80's and 90's when filenames were restricted to a maximum of 11 characters each. There were no standards for naming files at that time; each manufacturer had its own practice and the industry ended up with too many proprietary applications and operating nuances. Regardless, their early work became the foundation of the C37.232 standard and especially so because they had to create meaningful filenames using 11 characters only. The task was monumental and the ingenuity was amazing.

For example, one of the early manufacturers succeeded in including 25 characters of key information in the 11 filename character space. The key information included event date and time up to "12/31/2079-23:59:59.99" (which is 22 characters long) and the recorder number up to "255" (which is 3 characters long). The recorder number is coded in the first 2 characters of the filename, and the date and time are coded in the last 9 characters of the filename. The resulting filename is not friendly because it requires special decoding software to read it. For example, the filename "G30BQ1EF.063" decodes to "device number 163 on 09/18/1991 at 14:15:00.630". Other manufactures were able to do even more by also adding priority and event type information into the mix. A detailed example of a similar algorithm is provided in appendix A of the standard and a list of other types of naming practices is also provided.

THE STANDARD PRACTICE ADOPTED IN C37.232-2007

The standard naming practice defines a readable, comma delimited, text format. The fields for the file name are as follows and in order:

Start Date, Start Time, Time Code, Station Identifier, Device Identifier, Company Name

The above fields are called “required fields”. Additional fields may be added as needed by the user and are called “user fields”. The standard requires that the user fields follow directly after the required fields and in order as shown here:

, User-1, User-2, User-3, User-4, and so on. Extension

All required and user fields are separated by commas. Only one comma is used to separate between fields (trailing commas should not be used). The extension will always follow at the end as shown above. In order to conform to the standard, the required fields must appear in the filename in order as specified above.

LIMITATIONS

In addition to the limitations mentioned in the background section regarding length and allowable characters, additional limitations exist while trying to copy long filenames onto CD and floppy disks. Specifically, floppies are limited to a very small number of long filenames per disk and current CDs do not work with filenames that are more than 64 characters long. Users who deal with thousands of long filenames per folder have also reported problems with their zipping applications. However, given the rate of advancement in technology, it is highly possible that these limitations will soon vanish or become obsolete. The naming convention has been successfully tested under a variety of operating systems including, but not limited to, Windows, Linux, and Novell.

REQUIRED FIELDS

Start Date: The field is a formatted numeric field. The width is six digits: the first two digits are for the year, the second two digits are for the month, and the last two digits are for the day. For example, the code 010203 means the date is February 3rd, 2001. The standard defines the Start Date field as being the date of the first sample in the file. However, if the last character of the Time Code field is set to a “t”, then the Start Date is the date at which the first trigger occurred in the file.

Start Time: the field is defined in military time format and can be specified to the required precision. For example, 170215183222, 170215183, 17021518, 170215, and 1702 are all acceptable times. The standard defines the Start Time field as being the time of day of the first sample in the file. However, if the last character of the Time Code field is set to a “t”, then the Start Time is the time at which the first trigger occurred in the file.

Time Code: the field is restricted to a maximum of seven formatted characters. The first character is the sign and is followed by up to five characters indicating the time zone offset from universal time (UT). The format is up to two digits for the hours, followed by an optional letter “h” and two digits for the minutes. For example, the code +10h30 means the

time zone offset is 10 hours and 30 minutes ahead of universal time (UT). The last character in this field is an optional letter “t” used to indicate whether the date and time fields are referencing the first sample or the sample at which the trigger occurred.

Station, Device & Company Identifier: users can formulate their own unique codes for the station where the originating device is located, the originating device itself, and the company that operates the originating device. These fields are variable length fields and can contain letters, numbers and some punctuation marks. Characters that are disallowed are, ? “ / \ < > * | : (i.e. comma, question mark, quotation mark, forward slash, backward slash, less than, greater than, asterisk, pipe, and colon).

USER FIELDS

One or more optional fields are allowed, as needed, as long as there is room available in the filename. The fields are called user fields and can be used for comments or for any other purpose that is desired. Some users have reported placing a priority measure in one of the user fields which enabled the reviewers to sort by priority and quickly identify the worst case records. Other users have reported placing the fault type, distance and magnitude in the user fields which helped them trend circuit behavior. The user fields are variable length fields similar to the station, device and company identifier fields.

EXTENSION FIELD

The exact field definition is up to the user. However, the extensions: .DAT, .CFG, .HDR, and .INF are reserved by the IEEE C37.111 COMTRADE standard. There are no other reserved extensions. However, the previously noted restrictions on filename extensions (in the background section) should be observed. For example, users should avoid using the extensions “.EXE”, “.BAT”, “.SYS”, “.COM”, “.DLL”, or any other commonly known or used filename extensions while naming their TSD files.

EXAMPLES

The following is an example filename using only the required fields. The Start Time is given in millisecond and the extension is that of a COMTRADE configuration file:

000809, 175215183, -4, sta80, ben717, nyiso. cfg

The following is another example but with the Start Time given to a tenth of a second and with 3 additional user fields for duration (95.9 seconds) and type:

000809, 1752152, -4, sta80, ben717, nyiso, 000000, 0001359, uf. cfg

The following example includes an additional user field for comments:

000809, 1752152, -4, sta80, ben717, nyiso, 000000, 0001359, uf, critical frequency. cfg

Here is another example with 2 user fields showing geographic position information:

000809, 175215183, -4, sta80, ben717, nyiso, 04305n, 07767w. cfg

SHORT FILENAMES

Naming a file is very much like naming a chapter in a book or a section in a paper. The intent is to define a short term or phrase that describes the contents. Users should always strive to create filenames that are short and meaningful. The use of abbreviated fields is recommended and the use of too many user fields is discouraged. The following is an example of a short filename for a user generated file representing calculated swings in system frequency:

030914, 160404, -5t, sys, hz, sce. xls

Notice that in this case the station identifier is set to “sys” (short for power system), and the device Identifier is set to “Hz” (short for hertz). The example is acceptable and is in full compliance with the standard. The use of short names is very important especially for readability and exchange across various types of storage media, communication links, and operating systems. The standard strongly recommends that users aspire to create filenames that are always less than 64 characters long.

UNIQUENESS

Uniqueness depends on the choices made by the user. The names are absolutely guaranteed to be unique if geographic position information is included. However, users should assign their filenames such that the required fields alone can provide a sufficient guarantee that the filenames are unique.

UNIVERSALITY

The intent of the standard is to address the issues of naming TSD files. However, many users have reported using the same type of naming practice for naming other types of files. The nature of the format is self-readable, that by simply looking at the fields, one can decipher their actual meaning. Regardless, the standard recommends that the extensible markup language (XML) be used for defining fields. In this way, subsequent conventions can define the appropriate types of fields for the appropriate types of applications (users would simply develop the definitions in XML). Accordingly, the procedures defined by the standard are universally applicable to all of the commonly known file types out there. To that extent, the standard recommends that users avoid mixing multiple types of naming conventions within the same folder.

FUTURE TRENDS

Many applications can be realized given a unique, informative filing system. Possible applications include, but are not limited to, time line managers, TSD file managers, universal displays, automated fault analysis systems, national and international banks of TSD files, and so on. A number of these applications are now being developed and some of them, like the TSD file manager, have already been realized and are available on the market. Additional standards work is also being done by the PSRC H10 working group with the objective of defining a common practice for naming installed electronic devices. Their work will add more structure to the existing practice, thus enabling more applications and opening up new horizons. Among the main beneficiaries of these applications are the utilities, system operators, reliability commissions, manufacturers, and third party providers. We can now, collectively, begin to realize the benefits of sharing a common filing system for our TSD files: “take any file from any device anywhere on the planet and seamlessly file it like filing a book in a library”.

BIBLIOGRAPHY

This listing of books, articles & standards is provided as sources for additional information:

“Washington View: SA - Going the Way of the Dinosaur,” Davis, D. A., GPS World, V. 11, No. 6, June 2000, pages 16-20

"IEEE Standard Common Format for Transient Data Exchange (COMTRADE) for Power Systems," IEEE Standard P37.111, 1991, 1999

"Survey of File Naming Schemes,” A. Makki, Fault and Disturbance Conference, Georgia Tech, May 1998

“GPS Land Navigation,” Ferguson, Glassford Publishing, Boise, Idaho, 1997

"Overview of FAT, HPFS, and NTFS File Systems,” Win NT Resource Kit, Chp18, 1997

"Data Compression in Digital Systems," Huffman, Chapman & Hall, New York, 1997

"The Data Compression Book," Nelson, M&T Publishing, California, 1992

"Working with MS-DOS files," Microsoft User Guide & Reference MS-DOS 5.0, 1991