IEEE Std C37.111-1999
(Revision of
IEEE Std C37.111-1991)

IEEE Standard Common Format for Transient Data Exchange (COMTRADE) for Power Systems

Sponsor

Power Systems Relay Committee
of the
IEEE Power Engineering Society

Approved 18 March 1999

IEEE-SA Standards Board

Abstract: A common format for data files and exchange medium used for the interchange of various types of fault, test, or simulation data for electrical power systems is defined. Sources of transient data are described, and the case of diskettes as an exchange medium is recommended. Issues of sampling rates, filters, and sample rate conversions for transient data being exchanged are discussed. Files for data exchange are specified, as is the organization of the data. A sample file is given.
Keywords: configuration file, data file, header file, information file, transient data.

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Print:	ISBN 0-7381-1666-1	SH94743
PDF:	ISBN 0-7381-1667-X	SS94743

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Introduction

[This introduction is not part of IEEE Std C37.111-1999, IEEE Standard Common Format for Transient Data Exchange (COMTRADE) for Power Systems.]

The increasing use of digital technology in devices such as protection, oscillograph, measurement, and control apparatus in electric power substations has created the potential for accumulating large numbers of digital records of power system transient events. In addition to these sources of digital data, analog and digital power-system simulators may be used to generate digital records. The users of these records are faced with the problem of having to cope with different formats used by each system to generate, store, and transmit records.

Differences between IEEE Std C37.111-1999 and IEEE Std C37.111-1991

[These differences are not a part of IEEE C37.111-1999, IEEE Standard Common Format for Transient Data Exchange (COMTRADE) for Power Systems.]

a) The Header (.HDR) file is explicitly defined as optional.

b) The Configuration (.CFG) file has been modified. A field containing the COMTRADE standard revision year has been added to distinguish files made under this or future revisions of the standard. If this field is not present, the data is assumed to comply with C37.111-1991. A field for a Time Stamp Multiplication Factor has been added to meet the need for long duration files. To assist in conversion of the data, three new scaling fields (primary, secondary, and primary-secondary) are added, defining the instrument tranformer winding ratios, and whether the recorded data is scaled to reflect primary or secondary values. Configuration fields for Status (Digital) Channel Information have been expanded to five fields to allow the same level of definition as for analog channels. Line Frequency is now defined as a floating point field. Support for Event Triggered data has been added by the addition of a new mode for Sampling Rate Information when the sampling rate is variable. The Date/Time Stamps format has been modified with the day of month preceding the month entry field, and the year field now has all four numbers of the year. A requirement that at least one leading space shall be tolerated in the data fields, including those fields for which no data is available (previously specified as comma delimiters with no spaces in between) has also been added.

c) A new format for a binary data (.DAT) file has been specified and the requirement for a user-supplied conversion program has been eliminated.

d) A new optional Information file (.INF) has been added to provide for transmission of extra public and private information in computer-readable form.

e) All field descriptions are explicitly defined with respect to: criticality, format, type, minimum/maximum length, and minimum/maximum value.

Participants

At the time this standard was completed the Working Group of the revision of IEEE Std C37.111-1991, IEEE Standard Common Format for Transient Data Exchange (COMTRADE) for Power Systems, had the following membership:

Bob Ryan, *Chair*
Charles Shank, *Vice Chair*

Mark Adamiak
Kenneth Behrendt
Gabriel Benmouyal
Jim Brandt
Bob Burnett, Jr.
Jack Chadwick
George Chirco
J. Dix Fulton
Jay Gosalia
Jerry W. Hohn
Dennis K. Holstein
Ann Howard
Herbert Jacobi
G. Roger Jensen
Mladen Kzunovic
Volker Leitloff
Jeffrey L. McElray
Peter McLaren
Harish Mehta
R. Jay Murphy
Arun G. Phadke
Bruce A. Pickett
Frank Plumptre
Mohindar S. Sachdev
Veselin Skendzic
Larry Smith
John Sperr
Stanley Thompson
Hans Heining Triebs
Eric Udren
Michael A. Xavier, Jr.
Murty V. Yalla

Other individuals who have contributed to review and comments are:

Mike Hoffman
Jerry A. Jodice
George Schaeffer
James S. Thorp
Tom Walters
David Weinbach
David Wood
Michael A. Xavier, Sr.

Coordinators:

William Brenner
Gary R. Engmann
Mack W. Grady
Robert W. Ingham
Douglas. B. Seely
Eric Udren

The following members of the balloting committee voted on this standard:

Alexander P. Apostolov
John Appleyard
Michael P. Baldwin
Ernst A. Baumgartner
Robert W. Beckwith
Miroslav Begovic
Kenneth Behrendt
John Boyle
Barni Bozoki
James A. Bright
Bob Burnett, Jr.
Mark Carpenter
Carlos H. Castro
George Chirco
D. Mason Clark
Stephen P. Conrad
Douglas C. Dawson
Robert W. Dempsey
Edward F. Dowling
Walter Elmore
William E. Feero
J. Dix Fulton
Jonathan D. Gardell
Jeffrey G. Gilbert
Peter A. Goodwin
Willliam D. Greason
Edward A. Guro
Robert W. Haas
George S. Haralampu
Irwin O. Hasenwinkle
Roger A. Hedding
Charles F. Henville
Jerry W. Hohn
Dennis K. Holstein
Stanley H. Horowitz
John Horwath
James D. Huddleston, III
James W. Ingleson
David W. Jackson
Herbert Jacobi
G. Roger Jensen
Jerry A. Jodice
Kumul J. Khunkhun
Joseph L. Koepfinger
J. S. Komisarek
Gerald E. Lee
Volker Leitloff
William A. Maguire
Will J. Marsh Jr
John R. Matras
John E. McConnell
Michael J. McDonald
Jeffrey L. McElray
Peter McLaren
Michael Meisinger
Gary L. Michel
Daleep C. Mohla
Brian Mugalian
R. Jay Murphy
George R. Nail
Stig L. Nilsson
David Oka
Joseph C. Osterhout
Robert D. Pettigrew
Arun G. Phadke
Bruce A. Pickett
Alan C. Pierce
Frank Plumptre
Roger E. Ray
Jesus Martinez Rodriquez
Bob Ryan
Mohindar S. Sachdev
Miriam P. Sanders
Charles Shank
Tarlochan Sidhu
Mark Simon
Veselin Skendzic
Larry Smith
James E. Stephens
William M. Strang
Charles Sufana
Malcolm J. Swanson
Lewis Douglas Sweeney
John H. Taylor
Richard P. Taylor
John T. Tengdin
Stanley Thompson
James S. Thorp
Joe T. Uchiyama
Eric Udren
Arnold Vitols
John B. Waclo
Charles L. Wagner
William P. Waudby
Michael A. Xavier, Jr
Murty V. Yalla
David J. Zaprazny
Karl Zimmerman
John A. Zipp
Stan Zocholl

When the IEEE-SA Standards Board approved this standard on 18 March 1999, it had the following membership:

Contents

IEEE Standard Common Format for Transient Data Exchange (COMTRADE) for Power Systems

1. Overview

1.1 Scope

This standard defines a format for files containing transient waveform and event data collected from power systems or power system models. The format is intended to provide an easily interpretable form for use in exchanging data. As such, it does not make use of the economies available from data encoding and compression that proprietary formats depend upon for competitive advantage. The standard is for files stored on physical media such as digital hard drives and diskettes. It is not a standard for transferring data files over communication networks.

1.2 Purpose

This standard defines a common format for the data files and exchange medium needed for the interchange of various types of fault, test, and simulation data.

The rapid evolution and implementation of digital devices for fault and transient data recording and testing in the electric utility industry have generated the need for a standard format for the exchange of data. These data are being used with various devices to enhance and automate the analysis, testing, evaluation, and simulation of power systems and related protection schemes during fault and disturbance conditions. Since each source of data may use a different proprietary format, a common data format is necessary to facilitate the exchange of such data between applications. This will facilitate the use of proprietary data in diverse applications and allow users of one proprietary system to use digital data from other systems.

2. Files and data storage

2.1 Categories of files

Files stored on digital devices and media consist of bytes representing a combination of alphabetic, numeric, symbol, punctuation, and other formatting characters. Depending on the format, a byte, part of a byte, or more than one byte, may be represented by a letter, number, or symbol (e.g., "A," "3," or "+"). There are three general classes of files used on computer systems: executable files, text files, and data files. The use of the file determines the category.

2.1.1 Executable files

Executable files contain a sequence of instructions suitable for processing by a computer. Computer programs are stored as executable files (.EXE). COMTRADE does not define executable files.

2.1.2 Text files

Text files imply data in human-readable form. A text file may be used for control of a computer program if the format is rigidly specified. COMTRADE text files use the character representation specified in ANSI X3.4-1986 [B1].[1] This is often called "ASCII format" or "Text (.TXT) format" by word processor programs.

COMTRADE defines one freeform ASCII text file intended for strictly human interpretation, the Header file. COMTRADE also defines three files in which the format is rigidly controlled and which are both human- and computer-readable—the Configuration file and the Information file, and the ASCII form of the Data file.

Most word processors can save text files in two or more formats. The text format contains only the characters actually typed, including punctuation and standard formatting characters such as carriage return/line feed. Other formats contain special characters specific to the particular word processor being used. The text format shall be used for the text files in a COMTRADE record to eliminate word processor-specific characters or codes. Programs intended to read COMTRADE files only require use of the typed characters that most word processor programs can read or print.

If no command exists in the word processor to save the file in this format, an alternative method is to use the print functions to print the text to disk to create the file.

2.1.3 Data files

Data files may contain numerical data, text data, or both. The data may be stored in either binary or ASCII format. Fields within ASCII format data files use defined text separated by commas, or some other common delimiter. As such, they are both human- and machine-readable. Most word processors cannot format, read, or write data files in binary form. However, many spreadsheet and data processing programs can read binary data files, if the format is known. Binary numbers must be processed by application-specific software to be easily interpreted by humans. COMTRADE defines one binary file, the binary form of the data file. Binary data are generally used when large amounts of data are to be stored because this uses less storage space (e.g., three bytes of binary data can represent numbers from 0 to 16 777 215 whereas three bytes of ASCII data can represent the numbers from 0 to 999). ASCII numbers have the advantage of being interpreted by humans, and by standard computer hardware and software.

2.2 Critical/non-critical data

Some of the data in the configuration file are not absolutely necessary for reproduction of the sample data, and some variables provided for in the configuration file may not be relevant to a particular application. Such data may be described as non-critical and may be omitted. An example of such non-critical data is the recording device channel name. However, the position normally occupied by such variables must be maintained in order to maintain the integrity of the file. If data are described as non-critical in any clause of this

[1]The numbers in brackets preceded by the letter B correspond to those of the bibliography in Annex F.

standard, the position may be left empty and the corresponding data separator retained following the preceding data separator with no intervening characters or spaces. Any data that are necessary for reproduction of the sample data are termed critical. If such data are missing, the file may be unusable.

2.3 Data representations

Data are stored in files as series of binary digits or “bits.” Each bit can be either a 1 or a 0. The bits are organized in groups of eight bits called bytes. When a computer reads the data in a file, it reads the data as a series of bytes.

2.3.1 Binary data

The eight bits in a byte can be organized in 256 different combinations. They can be used, therefore, to represent the numbers from 0 to 255. If larger numbers are needed, several bytes can be used to represent a single number. For example, 2 bytes (16 bits) can represent the numbers from 0 to 65 535. When the bytes are interpreted in this fashion, they are known as binary data. Several different formats are in common use for storage of numeric data in binary form.

2.3.2 ASCII data

As an alternative to a byte representing the numbers 0 to 255, a byte can be used to represent 255 different symbols. ASCII is a standard code of symbols that match 127 of the combinations of eight binary bits. For example, the byte 01000001 represents an uppercase “A” while 01100001 represents a lowercase “a.” With 127 different combinations, it is possible to represent all of the keys on the keyboard plus many other special symbols. The remainder of the 256 combinations available from an eight-bit format are used for drawing and other special application characters. To represent a number in ASCII format requires one byte for each digit of the number.

2.4 Data field delimiters and lengths

Data fields within a file or within a subset of data in a file must be separated from the other data fields so that they may be extracted for reading or manipulation. For instance, written text uses a space as a word delimiter. Computer files use a variety of delimiters. In the binary form of COMTRADE data files, the only delimiter is a strict definition of the length and position of each data variable, and a byte count of the position within the file is necessary to determine the limit of any data entry. The ASCII files defined by COMTRADE use the comma and the carriage return/line feed as data separators. This permits the use of variable field lengths, but means that these characters cannot be used within any data entry. Leading spaces or zeroes are allowed in ASCII numeric fields providing the permitted maximum character count is not exceeded.

2.4.1 Carriage return/line feed delimiter <CR/LF>

COMTRADE uses the symbol <CR/LF> to represent a data separator terminating the end of a line or a set of data. The delimiter is the combination of two ASCII formatting characters:

CR = carriage return takes the cursor or insertion point back to the beginning of the current line.

LF = line feed moves the cursor or insertion point to a new line below the current line.

The symbols “<” and “>” surrounding the CR/LF are used to delineate the delimiter from the nearby text within the standard and are not part of the delimiter. In most present-day computer programming or application environments, the two-character combination is automatically generated when the return or “ENTER” key is pressed.

2.4.2 Comma delimiter

The comma is used as a delimiter for data entries within a sample in COMTRADE configuration (.CFG), information (.INF), and ASCII format data (.DAT) files.

2.4.3 Field lengths

Field lengths are specified for many alphabetical or numerical variables in the COMTRADE standard. These limitations were specified to simplify reading lines of data containing many variables. For integer numeric variables, the maximum field length is one character longer than required to hold the maximum value for that field. This extra character space is allowed for a leading minus for signed numbers and to allow the application of simple programming techniques that automatically print the leading space, even for unsigned numbers. The 6-character and 10-character field lengths for data, time stamps, and sample numbers used in earlier revisions of COMTRADE are retained for backward compatibility.

2.5 Floating point notation

Real numbers may be stored in several ways. Numbers of limited range can be entered as a numeric string with a decimal point. For larger or smaller numbers, any reasonable limit on string length leads to a loss of resolution. In such cases, it is desirable to store the number in a format allowing use of a representation of the significant digits (mantissa) and a multiplier (exponent) format. Spreadsheets and other mathematical programs often use floating point notation to represent such numbers. COMTRADE allows the use of floating point notation (Kreyszig [B7]) to represent real numbers for conversion factors in the .CFG file. The terms exponential notation or scientific notation are sometimes used for this form and interpretations of the form vary. Since programs designed to read COMTRADE files must be able to recognize and interpret numbers represented in this format, one single format is defined here. The numbers shall be interpreted and displayed as follows.

A signed floating point value consists of an optional sign (+ or –) and a series of decimal digits containing an optional decimal point, followed by an optional exponent field that contains the character “e” or “E” followed by an optionally signed (+ or –) integer exponent. The exponent is a factor of base 10, so 3E2 means 3 multiplied by 100 (10^2) or 300. Correct interpretation of negative numbers and negative exponents requires the inclusion of the negative sign. For positive numbers or exponents the sign is optional and is assumed if absent.

The format shall be written as:

[±]dd[.]dddd[E[±]ddd]

where

— Square brackets surround any optional item.
— “d” represents any numeral between 0 and 9.
— At least one numeral must appear in the field.
— If the decimal point appears, at least one numeral must appear to the left and right.
— The character “e” or “E” represents “exponential” with base 10.
— If the exponential sign appears, it must be followed by at least one numeral, the intervening plus/ minus sign is optional if positive, but must be “+” or “–” not “±.”
— The numeric value following “E” must be an integer.

Examples**:**

Acceptable

1E2 (= 100)

1.23E4 (= 12 300)

0.12345E-5 (= 0.0000012345)

–1.2345E2 (= –123.45)

Unacceptable

.123 (one numeral must precede decimal)

123E (at least one numeral must follow “E”)

±0.123E±4 (plus/minus signs make the value indeterminate)

0.123 E4 (space before “E” not allowed)

2.6 Methods of accessing data in files

The two different methods used to access text and data files are random access and sequential order.

2.6.1 Random access files

Data within random access files can be retrieved or stored in any random sequence. The access time for each record is independent of the location of the data. Each data field has a specific address that can be used for reading or writing. COMTRADE does not use random access files.

2.6.2 Sequential files

Sequential files are accessed by reading or writing each data field in sequence. Individual data fields have no specific address and their position in the file is relative to the other variables. The exact byte-count position in the file is dependent on the length of the preceding variables. COMTRADE uses sequential files.

2.7 Primary to secondary ratios

The devices used to measure and record events on a high voltage system are not capable of accepting the high voltage and high currents of the power system directly. These devices are built to accept inputs in more manageable and less dangerous levels, termed secondary quantities. Voltage transformers and current transformers are used to reduce the voltage and current signals on the power system to these lower values. The transformer ratios are chosen so that when the power system is running at the rated or nominal primary value, the secondary value is at the nominal secondary value. The ratio is specified in primary-secondary order, the convention being that the primary is closest to the source of power. Primary ratings are available for all common voltages and load values on the power system. Common values for the secondary values are in the region of 70 V line-to-ground, and 1 A or 5 A. IEEE Std C57.13–1993[B6] specifies the description and rating of these transformers.

Thus, for a current transformer applied to a feeder and rated at 800:5, the secondary current will be at the nominal 5 A value only when the primary load current is 800 A. Lower values of load result in correspondingly lower values of secondary current.

For three-phase applications, voltage transformers are normally rated in phase-to-phase voltage values rather than phase-to-ground. The output of a voltage transformer rated at 345 kV:120 V will be 120 V phase-to-phase (70 V phase-to-ground) only when the primary system phase-to-phase voltage is 345 kV. The term line-to-line is used interchangeably with phase-to-phase, and similarly line-to-ground instead of phase-to-ground.

3. COMTRADE Files

Each COMTRADE record has a set of up to four files associated with it (see Clause 2.). Each of the four files carries a different class of information. The four files are as follows:

a) Header;
b) Configuration;
c) Data; and
d) Information.

All files in the set must have the same filename, differing only by the extensions that indicate the type of files.

File names are in the form xxxxxxxx.yyy. The xxxxxxxx portion is the name used to identify the record (e.g., FAULT1 or TEST_2). The .yyy portion of the file name is used to identify the type of file and is known as the extension: .HDR for the header file, .CFG for the configuration file, .DAT for data file(s), and .INF for the information file. The file names must follow the IBM compatible DOS (MS-DOS Version 6 Operating System User's Guide [B8]) conventions for legal characters within the file names (e.g., periods and spaces are not allowed as part of the file name). The file names are limited to eight characters and extensions are limited to three characters.

3.1 Header file (xxxxxxxx.HDR)

The header file is an optional ASCII text file created by the originator of the COMTRADE data, typically through the use of a word processor program. The data is intended to be printed and read by the user. The creator of the header file can include any information in any order desired. Examples of information to include are given in subclause 4.1. The header file format is ASCII.

3.2 Configuration file (xxxxxxxx.CFG)

The configuration file is an ASCII text file intended to be read by a computer program and, therefore, must be saved in a specific format. The configuration file contains information needed by a computer program in order to properly interpret the data (.DAT) file. This information includes items such as sample rates, number of channels, line frequency, channel information, etc.

One field in the first line of the configuration file identifies the year of the COMTRADE standard revision with which the file complies (e.g., 1991, 1999, etc.). If this field is not present or it is empty, then the file is assumed to comply with the original issue of the standard (1991). The configuration file also contains a field that identifies whether the companion data file is stored in ASCII or binary format. Details of the exact content and format of the configuration file are given in Clause 5.

The configuration file can be created with a word processing program or by a computer program that creates the configuration file from the data that is the source of the transient record. If a word processor is used to create the configuration file, it must save the data in ASCII text file format.

3.3 Data file (xxxxxxxx.DAT)

The data file contains the value for each input channel for each sample in the record. The number stored for a sample is a scaled version of the value presented to the device that sampled the input waveform. The stored data may be either zero-based or it may have a zero offset. Zero-based data spans from a negative number to a positive number (e.g., –2000 to 2000). Zero-offset numbers are all positive with a positive number chosen to represent zero (e.g., 0 to 4000, with 2000 representing zero). Conversion factors specified in the configuration file define how to convert the data values to engineering units. The data file also contains a sequence number and time stamp for each set of samples.

In addition to data representing analog inputs, inputs that represent on/off signals are also frequently recorded. These are often referred to as digital inputs, digital channels, digital sub-channels, event inputs, logic inputs, binary inputs, contact inputs, or status inputs. In this standard, this type of input is referred to as a status input. The state of a status input is represented by a number "1" or "0" in the data file.

The data files may be in either ASCII or binary format—a field in the configuration files indicates which format is used. A detailed description of the data file format is given in Clause 6.

3.4 Information file (xxxxxxxx.INF)

The information file is an optional file containing extra information that, in addition to the information required for minimum application of the data set, file originators may wish to make available to users. The format provides for public information that any user can read and use, and private information that may be accessible only to users of a particular class or manufacturer. The information file will be described in detail in Clause 7.

4. Header file

The header file is an ASCII text file for the storage of supplementary narrative information, provided for the user to better understand the conditions of the transient record. The header file is not intended to be manipulated by an applications program.

4.1 Content

Examples of information that may be included in the header file are as follows:

a) Description of the power system prior to disturbance;
b) Name of the station;
c) Identification of the line, transformer, reactor, capacitor, or circuit breaker that experienced the transient;
d) Length of the faulted line;
e) Positive and zero-sequence resistance and reactance, capacitance;
f) Mutual coupling between parallel lines;
g) Locations and ratings of shunt reactors and series capacitors;
h) Nominal voltage ratings of transformer windings, especially the potential and current transformers;
i) Transformer power ratings and winding connections;
j) Parameters of the system behind the nodes where the data was recorded (equivalent positive- and zero-sequence impedance of the sources);

k) Description of how the data was obtained, whether it was obtained at a utility substation or by simulating a system condition on a computer program such as an electro-magnetic transient program (EMTP);
l) Description of the anti-aliasing filters used;
m) Description of analog mimic circuitry;
n) The phase sequencing of the inputs; and
o) Number of discs on which the record is stored.

4.2 Filenames

Header filenames shall have the .HDR extension to distinguish them from the configuration, data, and information files in the same set and to serve as a convention that is easy to remember and identify.

4.3 Format

The header file shall be a freeform ASCII text file of any length.

5. Configuration file

The configuration file is an ASCII text file that provides the information necessary for a human or a computer program to read and interpret the data values in the associated data files. The configuration file is in a predefined, standardized format so that a computer program does not have to be customized for each configuration file.

5.1 Content

The configuration file shall have the following information:

a) Station name, identification of the recording device, and COMTRADE Standard revision year;
b) Number and type of channels;
c) Channel names, units, and conversion factors;
d) Line frequency;
e) Sample rate(s) and number of samples at each rate;
f) Date and time of first data point;
g) Date and time of trigger point;
h) Data file type; and
i) Time Stamp Multiplication Factor.

5.2 Filenames

Configuration filenames shall have the .CFG extension to distinguish them from header, data, and information files in the same set and to serve as a convention that is easy to remember and identify.

5.3 Format

The configuration file is an ASCII text file in a standardized format. It must be included with every file set to define the format of the data file.

The file is divided into lines. Each line shall be terminated by a carriage return and line feed. Commas are used to separate fields within a line. The data separator comma is required even if no data is entered into a field. Since commas, carriage returns, and line feeds are used as data separators, they are not legal characters within any field. For example, a channel name such as “Pacific West, Line number two” shall be interpreted as two separate fields. The use of data separators allows the field length to be variable so that leading or padding zeroes, or spaces are not required. However, because some programming languages reserve a leading character position for a minus sign, programs intended to read COMTRADE files shall be written to tolerate at least one leading space in fields. The information in each line of the file must be listed in the exact order shown in subclauses 5.3.1–5.3.9. The lines must appear in the exact order shown in subclause 5.5. Deviations from this format will invalidate the file set.

5.3.1 Station name, identification, and revision year

The first line of the configuration file shall contain the station name, the recording device identification, and the COMTRADE Standard revision year.

station_name,rec_dev_id,rev_year <CR/LF>

where

station_name is the name of the substation location. Non-critical, alphanumeric, minimum length = 0 characters, maximum length = 64 characters.

rec_dev_id is the identification number or name of the recording device. Non-critical, alphanumeric, minimum length = 0 characters, maximum length = 64 characters.

rev_year is the year of the standard revision, e.g. 1999, that identifies the COMTRADE file version. Critical, numeric, minimum length = 4 characters, maximum length = 4 characters. This field shall identify that the file structure differs from the file structure requirement in the original IEEE Std C37.111-1991 COMTRADE Standard. Absence of the field or an empty field is interpreted to mean that the file complies with the 1991 version of the standard.

5.3.2 Number and type of channels

This statement contains the number and type of channels as they occur in each data record in the data file:

TT,##A,##D <CR/LF>

where

TT is the total number of channels. Critical, numeric, integer, minimum length = 1 character, maximum length = 7 characters, minimum value = 1, maximum value = 999999, TT must equal the sum of ##A and ##D below.

##A is the number of analog channels followed by identifier A. Critical, alphanumeric, minimum length = 2 characters, maximum length = 7 characters, minimum value = 0A, maximum value = 999999A.

##D is the number of status channels followed by identifier D. Critical, alphanumeric, minimum length = 2 characters, maximum length = 7 characters, minimum value = 0D, maximum value = 999999D.

5.3.3 Analog channel information

This group of lines contains analog channel information. There is one line for each analog channel, the total number of analog channel lines must equal ##A (see subclause 5.3.2). If the analog channel count = 0, then there are no analog channel information lines. The following format shall be used:

An,ch_id,ph,ccbm,uu,a,b,skew,min,max,primary,secondary,PS <CR/LF>

where

An is the analog channel index number. Critical, numeric, integer, minimum length = 1 character, maximum length = 6 characters, minimum value = 1, maximum value = 999999. Leading zeroes or spaces are not required. Sequential counter from 1 to total number of analog channels (##A) without regard to recording device channel number.

ch_id is the channel identifier. Non-critical, alphanumeric, minimum length = 0 characters, maximum length = 64 characters.

ph is the channel phase identification. Non-critical, alphanumeric, minimum length = 0 characters, maximum length = 2 characters.

ccbm is the circuit component being monitored. Non-critical, alphanumeric, minimum length = 0 characters, maximum length = 64 characters.

uu are the channel units (e.g., kV, V, kA, A). Critical, alphabetical, minimum length = 1 character, maximum length = 32 characters. Units of physical quantities shall use the standard nomenclature or abbreviations specified in IEEE Std 260.1–1993 [B4] or IEEE Std 280–1985 (R1996) [B5], if such standard nomenclature exists. Numerical multipliers shall not be included. Standard multiples such as k (thousands), m (one thousandth), M (millions), etc. may be used.

a is the channel multiplier. Critical, real, numeric, minimum length = 1 character, maximum length = 32 characters. Standard floating point notation may be used (Kreyszig [B7]).

b is the channel offset adder. Critical, real, numeric, minimum length = 1 character, maximum length = 32 characters. Standard floating point notation may be used (Kreyszig [B7]).

The channel conversion factor is ax+b. The stored data value of x, in the data (.DAT) file, corresponds to a sampled value of (ax+b) in units (uu) specified above. The rules of mathematical parsing are followed such that the data sample "x" is multiplied by the gain factor "a" and then the offset factor "b" is added. Manipulation of the data value by the conversion factor restores the original sampled values. See Annex E for example.

skew is the channel time skew (in μs) from start of sample period. Non-critical, real number, minimum length = 1 character, maximum length = 32 characters. Standard floating point notation may be used (Kreyszig [B7]).

The field provides information on time differences between sampling of channels within the sample period of a record. For example, in an eight-channel device with one A/D converter without synchronized sample and hold running at a 1 ms sample rate, the first sample will be, at the time, represented by the **timestamp**; the sample times for successive channels within each sample period could be up to 125 μs behind each other. In such cases the skew for successive channels will be 0; 125; 250; 375...; etc.

min is the range minimum data value (lower limit of possible data value range) for data values of this channel. Critical, integer, numeric, minimum length = 1 character, maximum length = 6 characters, minimum value = –99999, maximum value = 99999 (in binary data files the range of data values is limited to –32767 to 32767).

max is the range maximum data value (upper limit of possible data value range) for data values of this channel. Critical, integer, numeric, minimum length = 1 character, maximum length = 6 characters, minimum value = –99999, maximum value = 99999 (in binary data files the range of data values is limited to –32767 to 32767).

primary is the channel voltage or current transformer ratio primary factor. Critical, real, numeric, minimum length = 1 character, maximum length = 32 characters.

secondary is the channel voltage or current transformer ratio secondary factor. Critical, real, numeric, minimum length = 1 character, maximum length = 32 characters.

PS is the primary or secondary data scaling identifier. The character specifies whether the value received from the channel conversion factor equation ax+b will represent a primary (P) or secondary (S) value. Critical, alphabetical, minimum length = 1 character, maximum length = 1 character. The only valid characters are: p,P,s,S.

The data in the data file, the channel conversion factors, and the channel units can refer to either primary or secondary units. So, a 345 kV to 120 V transformer for a channel in which the units are kV will have the primary factor of 345 and a secondary factor of 0.12 (345, 0.12). The primary or secondary variable (PS) is provided as a means to calculate the equivalent primary or secondary values in applications where the primary or secondary value is desired and the alternate value is provided. If the data originate in an environment that has no primary/secondary relationship such as an analog power system simulator, the primary-secondary ratio shall be set to 1:1. With the determination of the primary (P) or secondary (S) values from the ax+b equation, the user can then determine the values required for analysis or playback.

	Setting of variable PS	
Value required	**P (provides primary values)**	**S (provides secondary values)**
Primary	Use value	Multiply by primary value and divide by secondary value
Secondary	Divide by primary value and multiply by secondary value	Use value

5.3.4 Status (digital) channel information

This group of lines contains the status channel information. There is one line for each status channel. The total number of status channel lines must equal ##D (See sublause 5.3.2). If the status channel count = 0, then there are no status channel information lines. The following format shall be used:

Dn,ch_id,ph,ccbm,y <CR/LF>

where

Dn is the status channel index number. Critical, integer, numeric, minimum length = 1 character, maximum length = 6 characters, minimum value = 1, maximum value = 999999. Leading zeroes or spaces are not required. Sequential counter ranging from 1 to total number of status channels (##D) without regard to recording device channel number.

ch_id is the channel name. Non-critical, alphanumeric, minimum length = 0 characters, maximum length = 64 characters.

ph is the channel phase identification. Non-critical, alphanumeric, minimum length = 0 characters, maximum length = 2 characters.

ccbm is the circuit component being monitored. Non-critical, alphanumeric, minimum length = 0 characters, maximum length = 64 characters.

y is the normal state of status channel (applies to status channels only), that is, the state of the input when the primary apparatus is in the steady state “in service” condition. Critical, integer, numeric, minimum length = 1 character, maximum length = 1 character, the only valid values are 0 or 1.

The normal state of status channel does not carry information regarding the physical representation of the status signal, whether there is a clean contact (open or closed) or a voltage (live or dead). The purpose is to define whether a 1 represents the normal or abnormal state.

5.3.5 Line frequency

The line frequency shall be listed on a separate line in the file:

lf <CR/LF>

where

lf is the nominal line frequency in Hz (for example, 50, 60, 33.333). Non-critical, real, numeric, minimum length = 0 characters, maximum length = 32 characters. Standard floating point notation may be used (Kreyszig [B7]).

5.3.6 Sampling rate information

This section contains information on the sample rates and the number of data samples at a given rate.

For files with one or multiple predetermined sample rates, the information comprises one line with the total number of sampling rates followed by a line for each sample rate including the number of the last sample at this sample rate. There shall be one line of sample rate and end sample number information for each sampling rate within the data file. For files with continuously variable sample periods, such as event-triggered files, the sample rate information comprises two lines: one line with a zero signifying that there are no fixed sample periods or rates, and a second line including a zero signifying that the sample period is not fixed and the number of the last sample in the data file.

Note that, if **nrates** and **samp** are zero, the **timestamp** in the data file becomes critical and **endsamp** must be set to the number of the last sample in the file. When both the **nrates** and **samp** variable information and the **timestamp** information is available, use of **nrates** and **samp** variables is preferred for precise timing

nrates <CR/LF>
samp,endsamp <CR/LF>

where

nrates is the number of sampling rates in the data file. Critical, integer, numeric, minimum length = 1 character, maximum length = 3 characters, minimum value = 0, maximum value = 999.

samp is the sample rate in Hertz (Hz). Critical, real, numeric, minimum length = 1 character, maximum length = 32 characters. Standard floating point notation may be used (Kreyszig [B7]).

endsamp is the last sample number at sample rate. Critical, integer, numeric, minimum length = 1 character, maximum length = 10 characters, minimum value = 1, maximum value = 9999999999.

5.3.7 Date/time stamps

There are to be two date/time stamps in the configuration file. The first one is for the time of the first data value in the data file. The second one is for the time of the trigger point. They shall be displayed in the following format:

dd/mm/yyyy,hh:mm:ss.ssssss <CR/LF>
dd/mm/yyyy,hh:mm:ss.ssssss <CR/LF>

where

dd is the day of month. Non-critical, integer, numeric, minimum length = 1 character, maximum length = 2 characters, minimum value = 1, maximum value = 31.

mm is the month. Non-critical, integer, numeric, minimum length = 1 character, maximum length = 2 characters, minimum value = 1, maximum value = 12.

yyyy is the year. Non-critical, integer, numeric, minimum length = 4 characters, maximum length = 4 characters, minimum value = 1900, maximum value = 9999. All 4 characters of the year shall be included.

The variables dd, mm, and yyyy are grouped together as one field, the numbers being separated by the “slash” character with no intervening spaces.

hh is the hour. Non-critical, integer, numeric, minimum length = 2 characters, maximum length = 2 characters, minimum value = 00, maximum value = 23. All times are to be shown in 24 h format.

mm are the minutes. Non-critical, integer, numeric, minimum length = 2 character, maximum length = 2 characters, minimum value = 00, maximum value = 59.

ss.ssssss are the seconds. Non-critical, decimal, numeric, resolution = 1 microsecond, minimum length = 9 characters, maximum length = 9 characters, minimum value = 00.000000, maximum value = 59.999999.

All values for the date and time are to be preceded and padded by zeros, as required. If any data for the time and date stamp is missing, field separator commas/<CR/LF> may follow each other without intervening characters, or the correctly formatted field may be filled with numeric values replaced by zeros.

5.3.8 Data file type

The data file type shall be identified as an ASCII or binary file by the file type identifier in the following format:

ft <CR/LF>

where

ft is the file type. Critical, alphabetical, non-case sensitive, minimum length = 5 characters, maximum length = 6 characters. Only text allowed = ASCII or ascii, BINARY or binary.

5.3.9 Time stamp multiplication factor

This field shall be used as a multiplication factor for the time stamp (**timestamp**) field in the data file(s) to allow for long duration recordings to be stored in COMTRADE format. The time stamp has a base unit of microseconds. The elapsed time from the first data sample in a data file to the sample marked by any time stamp field in that data file is the product of the time stamp for that data sample and the time multiplier in the configuration file (**timestamp** × **timemult**).

timemult <CR/LF>

where

timemult is the multiplication factor for the time differential (timestamp) field in the data file. Critical, real, numeric, minimum length = 1 character, maximum length = 32 characters. Standard floating point notation may be used (Kreyszig [B7]).

5.4 Missing data in configuration files

The configuration file format provides for the fact that some data may be unavailable. However, it is understood that lack of some critical data can make the file set unusable. Some data are therefore specified as non-critical and some as critical. Loss, or lack, of critical data in the configuration file renders the file set invalid and as not conforming to the standard. Loss, or lack, of non-critical data in the configuration file does not render the file non-conforming and does not make the file set unusable. When data are missing, the data separators follow each other with no intervening characters unless otherwise specified elsewhere in this clause. Programs intended to read COMTRADE files shall be written to tolerate data separators immediately following each other with no intervening spaces (null fields).

5.5 Configuration file layout

station_name,rec_dev_id,rev_year <CR/LF>
TT,##A,##D <CR/LF>
An,ch_id,ph,ccbm,uu,a,b,skew,min,max,primary,secondary,PS <CR/LF>
An,ch_id,ph,ccbm,uu,a,b,skew,min,max,primary,secondary,PS <CR/LF>
An,ch_id,ph,ccbm,uu,a,b,skew,min,max,primary,secondary,PS <CR/LF>
An,ch_id,ph,ccbm,uu,a,b,skew,min,max,primary,secondary,PS <CR/LF>
Dn,ch_id,ph,ccbm,y <CR/LF>
Dn,ch_id,ph,ccbm,y <CR/LF>
lf <CR/LF>
nrates <CR/LF>
samp,endsamp <CR/LF>
samp,endsamp <CR/LF>
dd/mm/yyyy,hh:mm:ss.ssssss <CR/LF>
dd/mm/yyyy,hh:mm:ss.ssssss <CR/LF>
ft <CR/LF>
timemult <CR/LF>

6. Data file

The data file contains the data values that are scaled representations of the sampled event. The data must conform exactly to the format defined in the configuration file so that the data can be read by a computer program. The data file type (**ft**) field defined in the configuration file specifies the file type. For binary data files **ft** is set to binary. For ASCII data files **ft** is set to ASCII.

6.1 Content

The data file contains the sample number, time stamp, and data values of each channel for each sample in the file. All data in data files are in integer format. In ASCII data files, the data for each channel within a sample are separated from the succeeding channel data by a comma. This is commonly called “comma delimited format.” Sequential samples are separated by a <CR/LF> between the last channel data value in a sample and the sample number of the succeeding sample. In binary files there are no separators between the data for each channel within a sample or between sequential sample periods. No other information is contained in the data file.

6.2 Data filenames

Data filenames shall have the .DAT extension to distinguish them from header, configuration, and information files in the same set and to serve as a convention that is easy to remember and identify. The filename itself shall be the same for header, configuration, data, and information files to associate all of the files.

In the event that the total storage space required for the file set exceeds 1.44 MB (the amount of data that will fit onto one double density 3.5 in floppy disk), the data file may be segmented into multiple files, each of less than 1.44 MB each. In this case, the last two characters of the .DAT extension shall be changed from AT to the sequence number of the file, maintaining the file extension length of three characters. This will allow for data files from .D00 to .D99, thus allowing a maximum of 100 data files.

6.3 ASCII data file format

The ASCII data file shall be divided into rows and columns. The number of data rows varies with the length of the recording and, thus, affects the length of the file. Each row shall be divided into TT+2 columns where TT is the total number of channels, analog and status, in the recording; and the other two are for the sample number and time stamp. The number of columns is dependent upon the recording system and also affects the file length. Field lengths specified for ASCII data files are maximum values and are not fixed lengths. All numeric characters, including sign notation, shall fit within the field length limits.

a) The first column contains the sample number.
b) The second column is the time stamp for the data of that sample number.
c) The third set of columns contain the data values that represent analog information.
d) The fourth set of columns contain the data for the status channels.

   If all the columns containing data values do not fit on the same line, they are continued *without* a carriage return/line feed (<CR/LF>) until all data values for that sample have been displayed. The last value shall be terminated with carriage return/line feed (<CR/LF>).
e) The next row (line) begins with the next sample number followed by the next data set.
f) An ASCII end of file (EOF) marker (“1A” HEX) shall be placed immediately following the carriage return/line feed (<CR/LF>) of the last data row of the file.

Each data sample record shall consist of integers arranged as follows:

$$\mathbf{n, timestamp, A_1, A_2, \cdots A_k, D_1, D_2, \cdots D_m}$$

where

n is the sample number. Critical, integer, numeric, minimum length = 1 character, maximum length = 10 characters, minimum value = 1, maximum value = 9999999999.

timestamp is the time stamp. Non-critical if **nrates** and **samp** variables in .CFG file are nonzero, critical if **nrates** and **samp** variables in .CFG file are zero. Integer, numeric, minimum length = 1 character, maximum length = 10 characters. Base unit of time is microseconds (μs). The elapsed time from the first data sample in a data file to the sample marked by any time stamp field is the product of the time stamp and the time multiplier in the configuration file (**timestamp** × **timemult**) in microseconds.
When both the **nrates** and **samp** variable information are available and the **timestamp** information is available, the use of **nrates** and **samp** variables is preferred for precise timing.

A_1 • • • A_k are the analog channel data values separated by commas until data for all analog channels are displayed. Non-critical, integer, numeric, minimum length = 1 character, maximum length = 6 characters, minimum value = –99999, maximum value = 99998. Missing analog values must be represented by placing the value 99999 in the field.

D_1 • • • D_m are the status channel data values separated by commas until data for all status channels are displayed. Non-critical, integer, numeric, minimum length = 1 character, maximum length = 1 character. The only valid values are 0 or 1. No provision is made for tagging missing status data and in such cases the field must be set to 1 or to 0. The last data value in a sample shall be terminated with carriage return/line feed.

6.4 Example ASCII data sample

Figure 1 shows an example data sample as specified in this standard. It has six analog values and six status values. It is taken from Annex C.

5, 667, –760, 1274, 72, 61, –140, –502,0,0,0,0,1,1 <CR/LF>

Figure 1—Example of data sample in ASCII format

6.5 Binary data files

The binary data files use the same basic structure as that used for the ASCII data files, with the exception that status channel data are compacted as described below. The format is sample number, time stamp, data value for each analog channel, and grouped status channel data for each sample in the file. No data separators are used; the data within a binary sample record is not separated by commas and the end of a sample record is not marked by carriage return/line feed characters. A binary data file is a continuous stream of binary data. Data translation is determined by sequential position within the file. If any data element is missing or corrupt, the sequence of variables will be lost and the file may be unusable. No provision is made for recovery under these circumstances.

Data are stored in binary format, but for convenience the values are shown in hexadecimal here. The data are not stored as ASCII representation of hexadecimal numbers. When storing a two-byte (16 bit) word, the standard DOS format is to store the least significant byte (LSB) of the data first, then the most significant byte (MSB). The two-byte data value **“1234”** will be stored in **“3412”** format. In storing a four-byte (32 bit) word, the least significant byte (LSB) of the word is stored first, then the next to least significant byte, then the next to most significant byte, then the most significant byte (MSB). The four-byte data value **“12345678”** will be stored in **“78563412”** format. The bits within a byte are numbered zero (least significant) to seven (most significant).

The sequential data in a binary data file represent the following:

a) Sample number and time stamp data are stored in unsigned binary form of four bytes each.
b) Analog channel sample data are stored in two's complement binary format of two bytes each. A data value of zero is stored as 0000 hexadecimal, –1 is recorded as FFFF. The maximum positive value is 7FFF, and the maximum negative value is 8001. The hexadecimal 8000 is reserved to mark missing data.
c) Status channel sample data are stored in groups of two bytes for each 16 status channels, with the least significant bit of a word assigned to the smallest input channel number belonging to that group of 16 channels. Thus, bit 0 of status word 1 (S1) is the status of digital input number 1, while bit 1 of status word 2 (S2) is the status of digital input number 18. No provision is made for marking missing status data, but a bit set to 1 or to 0 must be included to maintain the integrity of the word.

The length of the file will vary with the number of channels and the number of samples in the file. The number of bytes required for each sample in the file will be: $(Ak \times 2) + (2 \times INT(D_m/16)) + 4 + 4$.

Where

Ak	is the number of analog channels,
D_m	is the number of status channels,
$INT(D_m/16)$	is the number of status channel divided by 16 and rounded up to the next integer, and
4 + 4	represents 4 bytes each for the sample number and the time stamp.

Each data sample record shall consist of integers arranged as follows:

n timestamp A_1 A_2...A_k S_1 S_2...S_m

where

n is the sample number. Critical, integer, numeric, minimum length = 4 bytes, maximum length = 4 bytes, minimum value = 00000001, maximum value = FFFFFFFF.

timestamp is the time stamp. Non-critical if **nrates** and **samp** variables in .CFG file are nonzero, critical if **nrates** and **samp** variables in .CFG file are zero. Minimum length = 4 bytes, maximum length = 4 bytes, minimum value = 00000000, maximum value = FFFFFFFE.
Missing time stamp values shall be replaced by placing the value FFFFFFFF in the field to maintain the integrity of the file structure. Base unit of time is microseconds (μs). The elapsed time from the first data sample in a data file to the sample marked by any time stamp field is the product of the time stamp and the time multiplier in the configuration file (**timestamp** × **timemult**) in microseconds. When both the **nrates** and **samp** variable information and the **timestamp** information are available, the use of **nrates** and **samp** variables is preferred for precise timing.

$\left.\begin{array}{c} A_1 \\ \bullet \\ \bullet \\ \bullet \\ A_k \end{array}\right\}$ are the analog channel data values in two bytes continued until data for all analog channels are displayed. Non-critical, integer binary twos complement format, minimum length = 2 bytes, maximum length = 2 bytes, minimum value = 8001, maximum value = 7FFF. Missing analog values must be represented by placing the value 8000 in the field.

$\left.\begin{array}{c} S_1 \\ \bullet \\ \bullet \\ \bullet \\ S_m \end{array}\right\}$ is the status channel data values in 2 bytes (16 bits) for each 16 or part of 16 status channels continued until data for all status channels are displayed. Non-critical, integer unsigned binary format, minimum length = 2 bytes, maximum length = 2 bytes, minimum value = 0000, maximum value = FFFF. No provision is made for tagging missing status data and in such cases the bit may be set to 1 or to 0. However, to maintain the integrity of the word and the file, a 0 or 1 shall be stored for that bit.

If the number of status channels is not integrally divisible by 16, the higher channels shall be padded with 0 bits.

Example:

For a set of six status inputs as shown for the ASCII data file in subclause 6.4.(0,0,0,0,1,1),

a) Write these status inputs as a binary number (110000), recognizing that the channels are listed low bits first in the ASCII data file.
b) Then pad the number out to a 16 bit number (0000 0000 0011 0000).
c) Translate this to a hexadecimal value (00 30).
d) The data is then stored in LSB/MSB format (30 00).

6.6 Example binary data sample

Figure 2 shows an example data sample as specified in this standard. It has six analog values and six status values. It is the binary equivalent of the ASCII sample shown in subclause 6.4.

05 00 00 00 9B 02 00 00 08 FD FA 04 48 00 3D 00 74 FF 0A FE 30 00

Figure 2—Example of data sample in binary format

7. Information file

The information file (.INF) is an optional file. The .INF file provides for the exchange of information regarding the event recorded in the COMTRADE record that may enable enhanced manipulation or analysis of the data. This optional information is stored in a separate file to allow full backwards and forwards compatibility between current and future programs that utilize COMTRADE files. Any program reading data from information files shall be able to recognize any public section header, entry, or other data defined in this standard, and take any action in response to that data. There is a requirement that programs not recognizing certain data shall not alter that data in any way.

The file format is similar to the Windows™ ".INI" [B9] file format. Most programming languages now include functions for writing and reading from these files. Many programmers and users are familiar with the structure of these files.

Some of the sections in the information file duplicate information stored in the .CFG configuration file. The ".CFG" and ".DAT" files are the primary COMTRADE files and any data for which a variable is defined in either of these two files must be stored in the appropriate file even if duplicated in the .INF information file.

7.1 Content

The information file is an ASCII text file that is in a computer-readable specified format. The file contains information readable by the general user, and information specific to a given class of users, which may be unreadable to the general user. These two types of information are classed as public and private, respectively, and reside in separate sections of the file. Data stored in the information file shall be stored in a public section whenever a suitable section is defined. If a suitable predefined public section is not available, a private section may be used. The entries shall conform exactly to the format defined below so that the data can be read by a computer program.

7.2 Information file filenames

Information filenames shall have the ".INF" extension to distinguish them from header, configuration, and data files of the same set and to serve as a convention that is easy to remember and identify. The filename itself shall be the same as for the header, configuration, and data files with which it is associated.

7.3 Information file structure

The information file is divided into sections. Each section consists of a header line followed by a number of entry lines. There is no limit to the number of sections but there shall be at least one section per file. No data shall reside outside of a section. Each section is identified by a unique section header line. All data belong to the nearest section header above it in the file.

Generically the structure is as follows:

Public Record Information Section Header (information relating to the whole record)
    Publicly-Defined Record Information Entry Lines
Public Event Information Section Header (information relating to a particular channel and sample in the record)
    Publicly-Defined Event Information Entry Lines
Public File Description Section Header (information equivalent to .CFG file information relating to the whole record)
    Publicly-Defined File Description Entry Lines
Public Analog Channel #1 Section Header (information equivalent to .CFG file information relating to the first analog channel in the record)
    Publicly-Defined Analog Channel Entry Lines
Public Analog Channel #n Section Header (information relating to the next analog channel in the record, with a new section for each channel, up the number of analog channels in the record)
    Publicly-Defined Analog Channel Entry Lines
Public Status Channel #1 Section Header (information relating to the first status channel in the record)
    Publicly-Defined Status Channel Entry Lines
Public Status Channel #n Section Header (information relating to the next status channel in the record, with a new section for each channel, up the number of status channels in the record)
    Publicly-Defined Status Channel Entry Lines
Private Information Header
    Privately-Defined Record Information Entry Lines
Private Information Header
    Privately-Defined Record Information Entry Lines

7.3.1 Public sections

Public sections contain information in a form that can be used by equipment and/or software made by more than one manufacturer. Specific public section entry lines are defined in this document. Each revision of the standard will update public section variables and include any openly distributed private section entries in use at that time.

7.3.2 Private sections

Private sections contain manufacturer-specific information that is only useful with a specific vendor's software or hardware, or that is in a format unique to that manufacturer. Multiple private sections are allowed per manufacturer, and a single information file may contain private sections from several manufacturers. It is anticipated that manufacturers will generate private sections for specific purposes. If two or more manufacturers use similar private sections, a common form of the private section could be approved for use as public sections in future revisions of this standard.

7.4 File characteristics

Information files shall be in ASCII format as defined in subclause 2.1.2, with the following additional limitation:

a) Leading spaces are not allowed on any line;
b) File shall not include any user-added end of file (EOF) marker, such "1A" HEX; and
c) File length shall not exceed 64K.

7.5 Section headings

7.5.1 Public and private section header name formatting rules

The section name is delimited by square brackets. The section name resides alone on a line. No other data shall reside on the same line as the section name. The line is terminated with a <CR/LF>. The section name shall start with a letter character; a number or a symbol shall not be the first character of a section name. The section name must start with the word "Public" or, for private sections, a word clearly representing the organization to which the section belongs, followed by exactly one space, then followed by any number of words identifying the section. Individual words in proprietary company or organization names or trademarks comprising more than one word shall be concatenated by deleting the space between the words, or, to improve readability, by substituting the underline space character "_" for the space.

Section headings after the first section heading shall be separated from the preceding section header or entry lines by an empty line.

Public section headers shall be meaningful to a power systems engineer with limited computer knowledge.

7.5.2 Public section header naming examples

Examples:

Acceptable:

[Public File_Description] <CR/LF>

Unacceptable:

[Public DataSource]<CR/LF>	(Leading space)
[DataSource Public]<CR/LF>	(Must begin with word Public)

7.5.3 Private section header naming examples

Examples:

Acceptable:

[Company1 Input Ranges]<CR/LF>
[Company2 IsolatorType] <CR/LF>

Unacceptable:

[Company Name Input Ranges] <CR/LF>	(Spaces not allowed in owner identifier)
[12] <CR/LF>	(Starts with number)
{Bad Section}<CR/LF>	(Wrong bracket style)
[Bad Section<CR/LF>	(Missing bracket)
[Bad Section] Extra Data=Not Allowed<CR/LF>	(Extra text or entries on line after closing bracket)

7.6 Entry line

An entry line must start with one word 3–32 characters long followed by an equal (=) sign. The first word is the “Entry Name.” The entry name is a description of the function of the value string that follows. It is analogous to the name of a variable or constant in many programming languages. The entry name shall be meaningful when read in conjunction with the section name. The entry name need not be fully descriptive. The entry name can contain any printable characters with ASCII values between 33 and 127 decimal. The line shall be terminated with a <CR/LF>.

Examples:

Acceptable:

[Public File Description] <CR/LF>
Recording_Device_ID=Unit 123<CR/LF>

[Company2 Calibration] <CR/LF>
Ch1=2044.5, –7, 1<CR/LF>
Ch2=2046.2, 5.3, 1<CR/LF>
Ch3=2042.0, –0.4, –1<CR/LF>

Unacceptable:

[Company3 Calibration] <CR/LF>	
cl33421thvlst=2044.5,–7,1,2046.2,5.3,1,2042.0,–0.4,–1<CR/LF>	(Entry name not meaningful)
Ch 1= 2044.5, –7, 1<CR/LF>	(Extra spaces)
[Company3 Device Type] <CR/LF>	(No space between last entry in previous section)

7.6.1 Comment lines

An entry line prefixed with a semicolon is considered a comment line. Such lines are to be skipped by file reading algorithms and are used for comments or to comment out certain entries. The comment lines may be created by users or by a program. Comment lines shall not be used for extensive documentation or explanations, since this increases file size, file read time, and obscures the file structure to human readers.

When section headings are commented out, all entry lines in that section shall also be commented out. Failing to comment out the entry lines in a section where the heading has been commented out would cause any uncommented entry lines in that section to fall under the previous section heading.

Examples:

Acceptable:

[Company2 Calibration] <CR/LF>
; Sequence is gain, offset, polarity<CR/LF>
Ch1=2044.5, –7, 1<CR/LF>
Ch2=2046.2, 5.3, 1<CR/LF>
;Channel 2 replaced 7/16/95<CR/LF>
Ch3=2042.0, –0.4, –1<CR/LF>

Unacceptable:

;[Company3 Calibration] <CR/LF>	(Section heading commented out leaving orphan data)
Ch 1 = 2044.5, –7, 1<CR/LF>	(Extra spaces)
;This recorder uses 8 bit data and has 64 channels, test points on the card are high impedance and not galvanically isolated. <CR/LF>	(excessive and wrongly-placed documentation)

7.6.2 Value string

The value string is defined as all characters on an entry line from the equal sign to the end-of-line sequence. Value strings can contain one data item or several data items. Multiple data items are separated by commas. Numerical values shall begin immediately after the equal sign or comma delimiter with no leading space. Text strings that include a space after the equal sign or comma delimiter shall include the space as part of the value. For public sections, this information is specified in the standard. For private sections, the data type, format, and number of items per entry line are defined by the user.

7.7 Adding, modifying, and deleting information

Because several programs may write to, modify, and read from the .INF file independently, rules governing the deletion and addition of information are needed to reduce the potential for damage from programs operating without human intervention. Deliberate human intervention via user entry fields can be used to add or delete information from any section. However, this can render the information file unfit for the intended application.

7.7.1 Deleting information

A program cannot delete private sections that it did not create, nor may it modify or delete items from those sections. A program cannot delete public sections or items from those sections. However, items in public sections may be modified or items may be added.

7.7.2 Adding information

Any program may add entries to a public section. A program cannot add entries to a private section that it did not create. The format allows an unlimited number of public and private sections, each with an unlimited number of entries.

7.8 Public section header and entry line definitions

This standard specifies some public section headers and entry lines. If a publicly-defined section header is included, all of the defined entry lines for that section shall be included in the order listed. An entry line in which the equal sign "=" is followed by the line terminating <CR/LF> shall be interpreted as a null string (no characters) or a zero numeric value. If no suitable public format is available, new complementary private section definitions may be created restricting use to the originating manufacturer or user. Future revisions of this standard will document those in commonly accepted use at the time of the revision.

7.9 Public record information section

This public data section defines the software that writes the file, describes the COMTRADE event, and indicates the number of public event information sections included in the information file.

[Public Record_Information] <CR/LF>	(Section heading, must include brackets)
Source=Value<CR/LF>	
Record_Information=Value<CR/LF>	
Location=Value<CR/LF>	
max_current=Value<CR/LF>	(Entry lines)
min_current=Value<CR/LF>	
max_voltage=Value<CR/LF>	(Entry lines)
min_voltage=Value<CR/LF>	
EventNoteCount=Value<CR/LF>	

7.9.1 Section header definition

The following text string is publicly defined as a section heading for parameters applicable to the whole file:

[Public Record_Information]<CR/LF>

7.9.2 Public record information entry line definition

The following public record information entry lines and entry value variables are publicly defined:

Source=Value<CR/LF>

— An optional entry line providing a place for machine-readable text description of the software that was used to write the record. Value is an alphanumeric string with printable ASCII characters and white space; multiple entries are separated by commas. The string is the name and revision level of the program.

Record_Information=Value1,Value2,Value3,Value4<CR/LF>

— An optional entry line providing a place for machine-readable text description of the event. Value is an alphanumeric string with printable ASCII characters and white space; multiple entries are separated by commas for which the following values are publicly defined:

Value1: Fault, Unknown, Misoperation, Close, Trip, Reclose, Power Swing, Simulation.
Value2: AG, BG, CB, ABCG, AB, BC, CA, ABC, or any similar series of phase identifiers such as 12N, RS, etc.
Value3: Any other text string not being a variation of one of the above that helps describe the event.
Value4: Any other text string being an identifier for a unique device or type of device (e.g., transmission line, transformer).

Location=Value1,Value2<CR/LF>

— An optional entry for information regarding the location of the fault on a transmission line, if it is known. The following entries are publicly defined:

Value1: A real number representing distance to fault in terms of the following parameters.
Value2: Miles, kilometers, percent of line, percent of setting, Ohms.

max_current=Value<CR/LF>
min_current=Value<CR/LF>
max_voltage=Value<CR/LF>
min_voltage=Value<CR/LF>

— Optional entry lines for recorded minimum and maximum values of voltage and current for the record as a whole. The values are either primary or secondary values as specified by the PS variable in the channel definition using the unit specified in the .CFG file. They differ from the variables **min** and **max** in the .CFG file, which are the maximum possible range or physically limited values. Value is a real number corresponding to the highest (max_value) or lowest (min_value) value to be found in the data file after conversion by the appropriate channel scaling factors ax+b; (see subclause 5.3.3).

EventNoteCount=Value<CR/LF>

— An entry line for the number of Public Event Information sections in the .INF file. It is required only if Event Information sections are included. Value is an integer value equal to the total public event information in the information file. If this number is zero or if the EventNoteCount entry line does not exist, it is assumed that there are no public event information sections to be read.

7.10 Public event information definition

This public data section defines notes that are related to a specific event, sample, or channel within a COMTRADE record. This allows specific parts of the record to have data and descriptive text attached and later retrieved.

7.10.1 Section heading definition: [Public Event_Information_#n] <CR/LF>

The section heading is the string "Public Event_Information_#" with the information number "n" directly appended (no interposing space character allowed). The information number is a positive integer, starting at one, consecutive, and limited to the value of EventNoteCount in the Public Record Information section.

7.10.2 Public event information entry line definition

Channel_number=Value<CR/LF>
max_value=Value<CR/LF>
min_value=Value<CR/LF>
max_sample_number=Value<CR/LF>
min_sample_number=Value<CR/LF>
Sample_number_Text#=Value1,Value2<CR/LF>
Sample_number_Text#=Value1,Value2<CR/LF>

Data definition:

Where the "Sample_number" string appears in any of the following entries, Value or Value1 is the COMTRADE record sample number to which the information refers. The Sample_number is the ASCII integer number that will be stored in an ASCII data file; binary files sample numbers must be converted to ASCII integers before the match is made.

Channel_number

An entry line for the COMTRADE record channel number to which the information refers.

max_value and min_value

— Entry lines for recorded minimum and maximum values of voltage and current for the channel to which the information refers. The values are either primary or secondary values as specified by the PS variable in the channel definition using the unit specified in the .CFG file. They differ from the variables **min** and **max** in the .CFG file, which are the maximum possible range or physically limited values. Value is a real number corresponding to the highest (max_value) or lowest (min_value) value in the channel data after conversion by the appropriate channel scaling factors ax+b.

max_sample_number and min_sample_number

— Entry lines for the sample number at which the minimum or maximum recorded value occur.

Sample_number_Text#=Value1,Value2

— Entry lines for text notes on events. # is a sequential count of the number of Text entries, beginning at 1 and limited to 99 (2 characters); Value1 the sample number as described above; Value2 is any alphanumeric string with printable ASCII characters and white space. Hard returns (CR and/or LF) are considered terminating characters and are not allowed within the body of the string.

7.11 Public file description section

This public data section defines information that describes the record as a whole and is equivalent to data stored in the .CFG configuration file. The .CFG file is mandatory and the .CFG file containing the appropriate information shall be supplied, even if the configuration information is duplicated in the optional .INF file. This optional duplication of data permits users who use the .INF information file to access the data contained in the .CFG file without opening that file.

7.11.1 Section heading definition: [Public File_Description] <CR/LF>

The section heading is the string "Public File_Description" (no interposing space character allowed). Only one Public File_Description section is allowed per record. The entry lines duplicate the entry lines of the .CFG file, which define the record as a whole. Channel-specific definitions are contained in separate sections. If used, this section must contain an entry line equivalent to each line in the .CFG file except for analog and status channel definition lines. The entries for "Value" shall follow the rules for the equivalent data as specified in Clause 5.

7.11.2 Public file description entry line definition

Station_Name=Value
Recording_Device_ID=Value
Revision_Year=Value
Total_Channel_Count=Value
Analog_Channel_Count=Value
Status_Channel_Count=Value
Line_Frequency=Value
Sample_Rate_Count=Value
Sample_Rate_#1=Value

End_Sample_Rate_#1=Value
.
.
.
Sample_Rate_#n=Value
End_Sample_Rate_#n=Value
File_Start_Time=Value
Trigger_Time=Value
File_Type=Value
Time_Multiplier=Value

7.12 Public analog channel section

This public section defines entry variables for the analog channels of the record and provides information equivalent to that stored in the .CFG configuration file. The .CFG file is mandatory and a .CFG file containing the appropriate information shall be supplied even if the information is duplicated in the optional .INF file. This optional duplication of data permits users who use the .INF file access to the data contained in the .CFG file without opening that file.

7.12.1 Section heading definition: [Public Analog_Channel_#n]

The section heading is the string “Public Analog_Channel_#n” (no interposing space character allowed), where “n” is a number between 1 and the analog channel count for the record. One Public Channel Description section is required for each analog channel of the record. The entry lines duplicate information in the lines of the .CFG file, which pertain to individual analog channels. If used, this section shall contain an entry line for each variable on the analog channel line in the .CFG file. The entries for “Value” shall follow the rules for the equivalent variables as specified in Clause 5.

7.12.2 Public analog channel entry line definition

Channel_ID=Value
Phase_ID=Value
Monitored_Component=Value
Channel_Units=Value
Channel_Multiplier=Value
Channel_Offset=Value
Channel_Skew=Value
Range_Minimum_Limit_Value=Value
Range_Maximum_Limit_Value=Value
Channel_Ratio_Primary =Value
Channel_Ratio_Secondary=Value
Data_Primary_Secondary=Value

7.13 Public status channel section

This public section defines entry variables for the status channels of the record and provides information equivalent to that stored in the .CFG configuration file. The .CFG file is mandatory and a .CFG file containing the appropriate information shall be supplied even if the information is duplicated in the optional .INF file. This optional duplication of data permits users who use the .INF file to access the data contained in the .CFG file without opening that file.

7.13.1 Section heading definition: [Public Status_Channel_#n]

The section heading is the string “Public Status_Channel_#n” (no interposing space character allowed), where “n” is a number between 1 and the status channel count for the record. One Public Channel section is required for each status channel of the record. The entry lines duplicate information in the lines of the .CFG file, which deal with individual status channels. If used, this section shall contain an entry line for each variable on the status channel line in the .CFG file. The entries for “Value” shall follow the rules for the equivalent variables as specified in Clause 5.

7.13.2 Public status channel entry line definition

Channel_ID=Value
Phase_ID=Value
Monitored_Component=Value
Normal_State=Value

7.14 Sample .INF file

```
[Public Record_Information ]<CR/LF>
Source=COMwriter, V1.1<CR/LF>
Record_Information=Fault, AG, Trip,Transmission Line<CR/LF>
Location=189.2, miles<CR/LF>
max_current=3405.5<CR/LF>
min_current=–3087.2<CR/LF>
max_voltage=208.6<CR/LF>
min_voltage=–206.4<CR/LF>
EventNoteCount=2<CR/LF>
<CR/LF>
[Public Event_Information_#1] <CR/LF>
Channel_number=2<CR/LF>
max_value=204.5<CR/LF>
min_value=–205.1<CR/LF>
max_sample_number=168<CR/LF>
min_sample_number=15<CR/LF>
Sample_number_Text_#1=168,Transient on reclose<CR/LF>
Sample_number_Text_#2=15,Minimum during normal load <CR/LF>
<CR/LF>
[Public Event_Information_#2] <CR/LF>
Channel_number=1<CR/LF>
max_value=206.5<CR/LF>
min_value=205.1<CR/LF>
max_sample_number=159<CR/LF>
min_sample_number=9<CR/LF>
Sample_number_Text_#1=159,Transient on reclose<CR/LF>
Sample_number_Text_#2=9,Minimum during normal load <CR/LF>
<CR/LF>
[Public File_Description] <CR/LF>
Station_Name=Condie<CR/LF>
Recording_Device_ID=518<CR/LF>
Revision_Year=1999<CR/LF>
Total_Channel_Count=12<CR/LF>
Analog_Channel_Count=6<CR/LF>
Status_Channel_Count=6<CR/LF>
```

```
Line_Frequency=60<CR/LF>
Sample_Rate_Count=1<CR/LF>
Sample_Rate_#1=6000.000<CR/LF>
End_Sample_Rate_#1=885<CR/LF>
File_Start_Time=11/07/95,17:38:26.663700 <CR/LF>
Trigger_Time=11/07/95,17:38:26.687500 <CR/LF>
File_Type=ASCII <CR/LF>
Time_Multiplier=1<CR/LF>
<CR/LF>
[Public Analog_Channel_#1] <CR/LF>
Channel_ID=Popular Va-g<CR/LF>
Phase_ID=<CR/LF>
Monitored_Component=<CR/LF>
Channel_Units=kV<CR/LF>
Channel_Multiplier=0.14462<CR/LF>
Channel_Offset=0.0000000000<CR/LF>
Channel_Skew=0<CR/LF>
Range_Minimum_Limit_Value=–2048<CR/LF>
Range_Maximum_Limit_Value=2048<CR/LF>
Channel_Ratio_Primary =2000<CR/LF>
Channel_Ratio_Secondary=1<CR/LF>
Data_Primary_Secondary=P<CR/LF>
<CR/LF>
[Public Status_Channel_#1] <CR/LF>
Channel_ID=Va over<CR/LF>
Phase_ID=<CR/LF>
Monitored_Component=<CR/LF>
Normal_State=0<CR/LF>
<CR/LF>
[Company1 event_rec] <CR/LF>
recorder_type=1<CR/LF>
trig_set=0,0,0,0,6048,6272,0,0,0,0,0,0,0,0,0,0<CR/LF>
ch_type=1,1,1,1,1,1,1,0,0<CR/LF>
<CR/LF>
[Company1 analog_rec_#1] <CR/LF>
op_limit=15<CR/LF>
trg_over_val=f<CR/LF>
trg_under_val=f<CR/LF>
trg_roc=f<CR/LF>
inverted=0<CR/LF>
<CR/LF>
```


Annex A

(informative)

Sources and exchange media for transient data

There are several possible sources of transient data that could be converted to the COMTRADE standard for data exchange. Some examples are listed here.

A.1 Digital fault recorders

Digital fault recorders for monitoring power system voltages, currents, and events are supplied by several manufacturers. These devices record analog signals by periodically sampling them and converting the measured signals to digital values. Typical recorders monitor 16–128 analog channels and a comparable number of event (contact status) inputs. Sampling rates, analog-to-digital converter resolution, record format, and other parameters have not been standardized.

A.2 Analog tape recorders

Analog tape recorders record analog signals on magnetic tape, usually using frequency modulation techniques. Recorded tapes can be played back to drive oscilloscopes or plotters for visual examination of the recorded waveforms. Typical recorders monitor up to 32 analog signals.

By employing suitable hardware and software, the signals recorded on the analog tapes can be converted to digital records in any desired format. The fidelity of the resultant output is dependent upon the limitations of both the analog recorder and the digital conversion system. The loss in fidelity can be minimized by a proper choice of the sampling system.

A.3 Digital protective relays

New relay designs using microprocessors are currently being developed and marketed. Some of these relays have the ability to capture and store relay input signals in digital form and transmit this data to another device. In performing this function, they are similar to digital fault recorders, except that the nature of the recorded data may be influenced by the needs of the relaying algorithm. As with the digital fault recorders, record format and other parameters have not been standardized.

A.4 Transient simulation programs

Unlike the above devices that record actual power system events, transient simulation programs produce transient data by analyzing mathematical models of the power system. Because this analysis is carried out by a digital computer, the results are inherently in digital form suitable for digital data dissemination. While originally developed for the evaluation of transient overvoltage in power systems, these programs are finding increased usage in other types of studies, including test cases for digital relaying algorithms. Because of the ease with which the input conditions of the study can be changed, transient simulation programs can provide many different test cases for a relay.

A.5 Analog simulators

Analog simulators model power system operations and transient phenomena with scaled values of resistance, inductance, and capacitance, while operating at greatly reduced values of voltage and current. The components usually are organized with similar line segments that can be connected to form longer lines. The frequency response of the analog simulator primarily is limited by the equivalent length of the model segment and typically ranges from 1–5 kHz. As with the output of analog tape recorders, the analog output of the simulator could be converted to digital records with appropriate filtering and sampling.

A.6 Data exchange medium

A.6.1 Introduction

Electric power utilities record fault data for post-fault analysis to determine the nature and location of the fault, and to store a record for future use. The data are generally stored as oscillograms on magnetic tapes or paper or in computer data files. An oscillogram contains voltage and current waveforms that can be examined and analyzed. Digital computers cannot record voltage and current waveforms directly. The waveforms are quantified for storage in computer files. More recently, personal computers have been used to record fault data on diskettes and cassettes.

It is not convenient to transport magnetic tapes that are used with mainframe computers in the form of reel-to-reel or cassettes between utilities and individual users. This is especially true if the users are separated by long distances or are located in different countries. Also, the recipient of a magnetic tape must have a computer system compatible with the system on which the tape was prepared. It is more convenient to transport cassettes than to transport magnetic tapes. However, transferring data to and from cassettes is a slow process.

A.6.2 Recommended medium

The most commonly used computer systems today are personal computers equipped with floppy disk drives. The double-sided, high-density 1.44 MB 3.5 in floppy diskettes are the most popular form of file transfer. They can be placed easily in padded envelopes and sent by mail from one location to another. Because of the widespread use and the convenience with which they can be sent to another location by mail, it is recommended that 1.44 MB 3.5 in floppy diskettes be used for exchanging fault data. However, individuals exchanging data may use other mutually acceptable media.

Annex B

(informative)

Data exchange sampling rates

B.1 Introduction

This annex is concerned with issues of sampling rates, filters, and sample rate conversions for transient data being exchanged. Of special concern is the case in which data are captured at a high sample rate but a lower sample rate is required by the device or software using the data. The simple expedient of dropping every n^{th} sample is **not** the correct way of making the conversion. This section discusses the correct way to perform this common function, as well as other related topics.

Since it is difficult to anticipate all future uses of such standard test cases (e.g., future algorithms, architectures, microprocessors), it seems clear that high accuracy and high sampling rates are desirable in the test cases. Although many existing digital relays use 12 bit accuracy, 16 bit or higher resolution A/D converters may be used in the near future.

The sampling rate issue is similar. Samples obtained at a sampling frequency of 240 Hz, for example, must be obtained using a filter with a cutoff frequency of 120 Hz to avoid aliasing. It is straightforward to convert these samples to samples at higher sampling frequencies, but the effect of the anti-aliasing filter cannot be removed. It is possible to obtain samples at 960 Hz equivalent to the output of the 120 Hz anti-aliasing filter, but it is not possible to obtain samples at 960 Hz of the original (unfiltered) signal.

B.2 Sampling process structure

It is recommended that the original samples be obtained (after a proper anti-aliasing filter is used, if necessary) at as high an accuracy and as high a sampling rate as possible in a given installation. However, specific choices of sampling rates (see sampling rates in Tables B.1 and B.2) could make further use of the data much easier. Consider data obtained at a sampling rate of f_s Hz. It would be most convenient if there were a standard technique to convert from the data at f_s Hz to data that would have been obtained by the user's proposed system shown in Figure B.1.

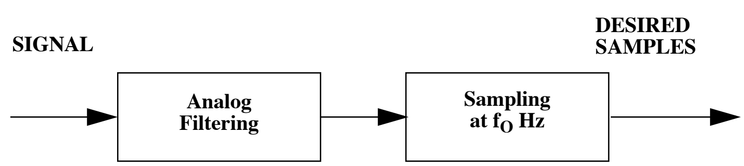

Figure B.1—Typical signal processing

Developments in digital signal processing present an efficient solution to the problem if there are integers L and M such that

$$Lf_s = Mf_o = f_{\mathrm{LCM}} \tag{B.1}$$

where

f_{LCM} is the least common multiple. The solution is shown in Figure B.2.

The box labeled FIR in Figure B.2 is a finite impulse response equivalent of the analog filter shown in Figure B.1 at a sampling rate of Lf_s Hz. Equation (B.1) is the key to the solution, and it limits sampling rates to some extent.

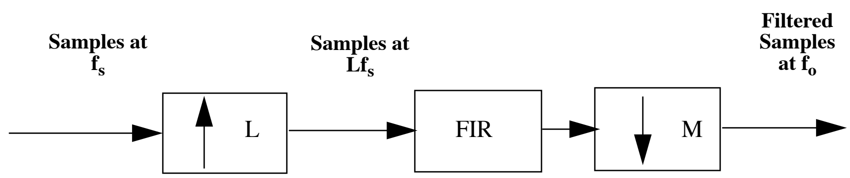

Figure B.2—DSP solution

The process of converting from samples at frequency f_s to samples at frequency f_0 is to determine the frequency f_{LCM} such that equation (B.1) is satisfied, provide an FIR description of the desired analog filter, and implement Figure B.2. The FIR description of the analog filter is a table of numbers corresponding to a digital filter description at the sampling frequency f_{LCM}. A standard technique for the FIR design might be to use an impulse equivalent filter where the n^{th} entry in the table was the impulse response of the analog filter at the n^{th} sample time. Other FIR filter design programs are available (Programs for Digital Signal Processing [B10]) and Annex D contains a program that implements Figure B.2

The transient response of the FIR filter at the beginning of the data must also be considered. If the FIR duration is one period of the nominal power system frequency, then a total of two periods of prefault data shall be included in the standard cases. Artificial prefault data can be supplied if it is not present. The FORTRAN program CONVERT (see Annex D) is an implementation of Figure B.2 that is an alternative to the program in Programs for Digital Signal Processing [B10]. The program is an illustration of the impulse invariant FIR filter for a second-order low-pass filter. Figure B.3 shows the output samples at 720 Hz with an input sampled at 4320 Hz.

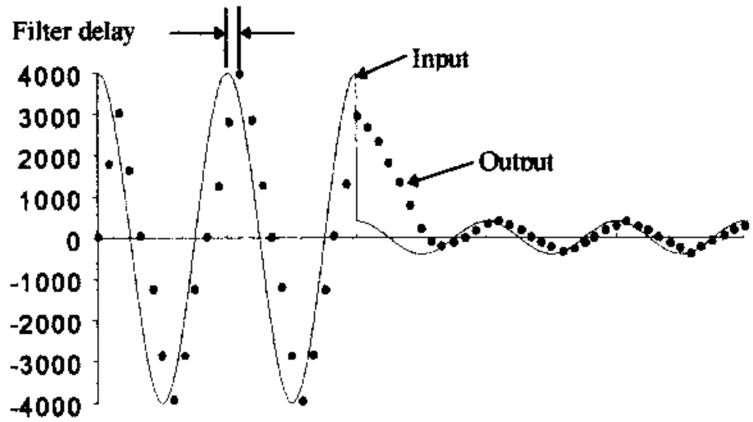

Figure B.3—Example of sample rate conversion

Table B.1—Frequencies corresponding to (f_{LCM} = 384 × f_{base}) samples/cycle

Samples/cycle	*f* for 60 Hz	*f* for 50 Hz
384	23 040	19 200
192	11 520	9600
128	7680	6400
96	5760	4800
64	3840	3200
48	2880	2400
32	1920	1600
24	1440	1200
16	960	800
12	720	600
8	480	400
6	360	300
4	240	200

Table B.2—Frequencies corresponding to (f_{LCM} = 3200 × f_{base}) samples/cycle

Samples/cycle	*f* for 60 Hz	*f* for 50 Hz
3200	192 000	160 000
1600	96 000	80 000
800	48 000	40 000
640	38 400	32 000
400	24 000	20 000
320	19 200	16 000
200	12 000	10 000
160	9600	8000
128	7680	6400
100	6000	5000
80	4800	4000
64	3840	3200
50	3000	2500
40	2400	2000
32	1920	1600
20	1200	1000
16	960	800
10	600	500
8	480	400
4	240	200

NOTE—The higher sampling frequencies in Tables B.1 and B.2 are artifacts of the technique of sampling rate conversion and data sharing. It is not intended that the high sampling rates be used to capture traveling wave phenomena. It is expected that the lower frequencies in Tables B.1 and B.2 will be the norm.

A further simplification would result if a single f_{LCM} were specified. The simplification would be that the user would have to specify a single FIR representation of the desired analog filtering at the specified f_{LCM}. Unfortunately, a single f_{LCM} that would satisfy all the sampling rates known to the working group would be so large as to make the description of an FIR filter unwieldy. The solution is to use two different common multiple frequencies f^1_{LCM} and f^2_{LCM}. Each frequency would produce a short list of sampling frequencies corresponding to an integer number of samples per cycle at the nominal power system frequency. Conversions between frequencies in a single list would be particularly simple. Conversions between frequencies that are not in a single list would require that the user determine the appropriate f_{LCM} for the application and then follow the same procedure. The two lists of recommended sampling frequencies are shown in Tables B.1 and B.2 for both 50 Hz and 60 Hz fundamental frequencies. It is assumed that the sampling frequencies are independent of the actual power system frequency and that the columns “samples per cycle” in Tables B.1 and B.2 are interpreted as the number of samples per cycle at the nominal power system frequency of 50 or 60 Hz.

B.3 Interpolation

The preceding subclause is based on the assumption that the original data consist of the samples taken directly after a properly designed anti-aliasing filter. The possibility that the data to be shared has been processed digitally must also be considered. If the digital processing can be represented as a linear shift-invariant operation that preserves the original sampling rate of f_s Hz, then it is straightforward to invert the digital processing.

As an example, let the original samples be the sequence $x(n)$ and assume that the average over the first four samples is used to produce the sequence $y(n)$,

$$y(n) = 1/4\ [x(n) + x(n-1) + x(n-2) + x(n-3)] \tag{B.2}$$

Given the sequence $y(n)$, it is possible to recover $x(n)$ with

$$x(n) = 4y(n) - x(n-1) - x(n-2) - x(n-3) \tag{B.3}$$

A more serious problem is encountered if decimation is involved in the digital processing, i.e., samples are eliminated and data is produced at a lower sampling rate. In the previous example, this might correspond to sharing only every fourth sample of $y(n)$ to form

$$z(n) = y(4n) \tag{B.4}$$

Programs for Digital Signal Processing [B10] shows a program for least-squares interpolation, i.e., to recover the missing samples from the sequence $y(n)$. It assumes, however, that the sequence $y(n)$ is band-limited to a bandwidth consistent with the lower sampling rate. If the digital filtering has effectively reduced the bandwidth, then the interpolation should be successful. The digital filtering (averaging) provided by Equation (B.2) might be acceptable; and, in time-critical applications, might be the only practical technique that can be used. In the absence of appropriate digital filtering, however, decimation introduces aliasing. In the previous example, if every fourth sample of the original sequence $x(n)$ is retained, this corresponds to sampling the original signal at $f_s/4$ Hz, but with an anti-aliasing filter with too large a bandwidth. The non-fundamental frequencies present in the waveforms will be distorted by aliasing. It is recommended that decimation be avoided, if possible, and that it only be used after appropriate analog or digital filtering.

Annex C

(informative)

Sample file

This annex includes copies of the files associated with a COMTRADE event such as might be recorded at a utility substation: the header, the configuration, and the data file in both ASCII and binary forms, and the information file. The header (SAMPLE.HDR), the configuration (SAMPLE.CFG), and the information (SAMPLE.INF) files are alphanumeric. The data file (SAMPLE.DAT) contains numerical information. Although both ASCII and binary forms of the data file are shown here, in practice only one data file can be associated with any given configuration file. The configuration file shown here specifies that the associated data file is in ASCII. If the binary file format were specified, the line of the configuration file which, in the example, reads “ASCII” would read “binary.”

C.1 SAMPLE.HDR

Currents, voltages, and digital outputs in this file were sampled from the Condie terminal of the 230 kV transmission line number 907, from Condie to Popular River. The 230 kV transmission line branches into a tee at the Condie end. On each side of the branch is a circuit breaker. The currents in the two branches are sampled and the sum of the currents in the two branches (i.e., current in the line) is also sampled.

The fault type and location are not known. The parameters of the system element on which the fault was experienced and the source impedances, therefore, are not known.

The operating conditions that existed immediately prior to the occurrence of the disturbance were not recorded. However, six cycles of pre-disturbance data are recorded in this file and the operating conditions can be calculated from that data.

The disturbance occurred on 11 July 1995 at 17:38:26.687500 hours.

Six cycles of pre-transient data and eight cycles of post-transient data are on the file. In total, there are fourteen cycles of data recorded on the file.

Data samples have been obtained at 6000 Hz. Anti-aliasing filters used for recording this data were second-order Butterworth filters that have a cutoff frequency of 2000 Hz.

The time skew of recording within each data set is zero. The nature of data in each column and the scaling factor for each operating parameter are as defined in the configuration file.

C.2 SAMPLE.CFG

```
Condie,518,1999 <CR/LF>
12,6A,6D <CR/LF>
1,Popular Va-g,,,kV, 0.14462,0.0000000000,0,–2048,2047,2000,1,P <CR/LF>
2,Popular Vc-g,,,kV, 0.14462,0.0000000000,0,–2048,2047,2000,1,P <CR/LF>
3,Popular Vb-g,,,KV, 0.14462,0.0000000000,0,–2048,2047,2000,1,P <CR/LF>
4,Popular Ia,,,A,11.5093049423,0.0000000000,0,–2048,2047,1200,5,P <CR/LF>
5,Popular Ib,,,A,11.5093049423,0.0000000000,0,–2048,2047,1200,5,P <CR/LF>
6,Popular Ic,,,A,11.5093049423,0.0000000000,0,–2048,2047,1200,5,P <CR/LF>
```



```
1,Va over,,,0 <CR/LF>
2,Vb over,,,0 <CR/LF>
3,Vc over,,,0 <CR/LF>
4,Ia over,,,0 <CR/LF>
5,Ib over,,,0 <CR/LF>
6,Ic over,,,0 <CR/LF>
60 <CR/LF>
1 <CR/LF>
6000.000,885 <CR/LF>
11/07/1995,17:38:26.663700 <CR/LF>
11/07/1995,17:38:26.687500 <CR/LF>
ASCII <CR/LF>
1
```

C.3 ASCII SAMPLE.DAT

```
  1,      0, –994, 1205,  100,   29, –135, –197,0,0,0,0,0,0 <CR/LF>
  2,    167, –943, 1231,   94,   37, –137, –275,0,0,0,0,0,0 <CR/LF>
  3,    333, –886, 1251,   87,   45, –139, –351,0,0,0,0,0,1 <CR/LF>
  4,    500, –826, 1265,   80,   52, –140, –426,0,0,0,0,1,0 <CR/LF>
  5,    667, –760, 1274,   72,   61, –140, –502,0,0,0,0,1,1 <CR/LF>
  6,    833, –689, 1279,   64,   68, –140, –577,0,0,0,0,0,0 <CR/LF>
  7,   1000, –613, 1279,   56,   76, –139, –651,0,0,0,0,0,0 <CR/LF>
  8,   1167, –537, 1275,   48,   83, –139, –723,0,0,0,0,0,0 <CR/LF>
  ...
  ...
883, 147000,  394, –446,   –1,    0,   –1, –345,0,0,0,0,0,0 <CR/LF>
884, 147167,  378, –417,   –2,    0,   –1, –366,0,0,0,0,0,0 <CR/LF>
885, 147333,  360, –387,   –2,    0,   –1, –385,0,0,0,0,0,0 <CR/LF>
<1A><CR/LF>
```

C.4 Binary SAMPLE.DAT

NOTE—The sample file is shown in HEX DUMP format, as it will be shown if viewed by a typical binary file viewer. The spaces between the bytes and the number of characters on a line are a function of the program used. The four byte sample numbers have been put in **BOLD** font manually, to aid in reading the file fragment.

01 00 00 00 00 00 00 00 1E FC B5 04 64 00 1D 00 79 FF 3B FF
00 00 **02 00 00 00** A7 00 00 00 51 FC CF 04 5E 00 25 00 77 FF
ED FE 00 00 **03 00 00 00** 4E 01 00 00 8A FC E3 04 57 00 2D 00
75 FF A1 FE 20 00 **04 00 00 00** F5 01 00 00 C6 FC F1 04 50 00
34 00 74 FF 56 FE 10 00 **05 00 00 00** 9C 02 00 00 08 FD FA 04
48 00 3D 00 74 FF 0A FE 30 00 **06 00 00 00** 43 03 00 00 4F FD
FF 04 40 00 44 00 74 FF BF FD 00 00 **07 00 00 00** EA 03 00 00
9B FD FF 04 38 00 4C 00 75 FF 75 FD 00 00 **08 00 00 00** 91 04
00 00 E7 FD FB 04 30 00 53 00 75 FF 2D FD 00 00 ...

... **73 0C 00 00** 38 3E 00 00 8A 01 42 FE FF FF 00 00 FF FF
A7 FE 00 00 **74 03 00 00** DF 3E 00 00 7A 01 5F FE FE FF 00 00
FF FF 92 FE 00 00 **75 03 00 00** 85 3F 00 00 68 01 7D FE FE FF
00 00 FF FF 7F FE 00 00

C.5 SAMPLE.INF

```
[Public Record_Information ] <CR/LF>
Source=COMwriter, v1.0<CR/LF>
Record_Information=Fault, AG, Trip, Transmission Line<CR/LF>
Location=189.2, miles<CR/LF>
max_current=3405.5<CR/LF>
min_current=–3087.2<CR/LF>
max_voltage=208.6<CR/LF>
min_voltage=–206.4<CR/LF>
EventNoteCount=2<CR/LF>
<CR/LF>
[Public Event_Information_#1] <CR/LF>
Channel_number=4<CR/LF>
max_value=504.5<CR/LF>
min_value=405.1<CR/LF>
max_sample_number=168<CR/LF>
min_sample_number=15<CR/LF>
Sample_number_Text_#1=168,Transient on reclose<CR/LF>
Sample_number_Text_#2=15,maximum on normal load <CR/LF>
<CR/LF>
[Public Event_Information_#2] <CR/LF>
Channel_number=5<CR/LF>
max_value=406.5<CR/LF>
min_value=405.1<CR/LF>
max_sample_number=159<CR/LF>
min_sample_number=9<CR/LF>
Sample_number_Text_#1=159,Transient on reclose<CR/LF>
Sample_number_Text_#2=9,maximum on normal load <CR/LF>
<CR/LF>
```

```
[Public File_Description] <CR/LF>
Station_Name=Condie<CR/LF>
Recording_Device_ID=518<CR/LF>
Revision_Year=1999<CR/LF>
Total_Channel_Count=12<CR/LF>
Analog_Channel_Count=6<CR/LF>
Status_Channel_Count=6<CR/LF>
Line_Frequency=60<CR/LF>
Sample_Rate_Count=1<CR/LF>
Sample_Rate_#1=6000.000<CR/LF>
End_Sample_Rate_#1=885<CR/LF>
File_Start_Time=11/07/95,17:38:26.663700 <CR/LF>
Trigger_Time=11/07/95,17:38:26.687500 <CR/LF>
File_Type=ASCII <CR/LF>
Time_Multiplier=1<CR/LF>
<CR/LF>
[Public Analog_Channel_#1] <CR/LF>
Channel_ID=Popular Va-g<CR/LF>
Phase_ID=<CR/LF>
Monitored_Component=<CR/LF>
Channel_Units=kV<CR/LF>
Channel_Multiplier=0.14462<CR/LF>
Channel_Offset=0.0000000000<CR/LF>
Channel_Skew=0<CR/LF>
Range_Minimum_Limit_Value=–2048<CR/LF>
Range_Maximum_Limit_Value=2047<CR/LF>
Channel_Ratio_Primary =2000<CR/LF>
Channel_Ratio_Secondary=1<CR/LF>
Data_Primary_Secondary=P<CR/LF>
<CR/LF>
[Public Analog_Channel_#2] <CR/LF>
Channel_ID=Popular Vc-g<CR/LF>
Phase_ID=<CR/LF>
Monitored_Component=<CR/LF>
Channel_Units=kV<CR/LF>
Channel_Multiplier=0.14462<CR/LF>
Channel_Offset=0.0000000000<CR/LF>
Channel_Skew=0<CR/LF>
Range_Minimum_Limit_Value=–2048<CR/LF>
Range_Maximum_Limit_Value=2047<CR/LF>
Channel_Ratio_Primary =2000<CR/LF>
Channel_Ratio_Secondary=1<CR/LF>
Data_Primary_Secondary=P<CR/LF>
<CR/LF>
[Public Analog_Channel_#3] <CR/LF>
Channel_ID=Popular Vb-g<CR/LF>
Phase_ID=<CR/LF>
Monitored_Component=<CR/LF>
Channel_Units=kV<CR/LF>
Channel_Multiplier=0.14462<CR/LF>
Channel_Offset=0.0000000000<CR/LF>
Channel_Skew=0<CR/LF>
Range_Minimum_Limit_Value=–2048<CR/LF>
Range_Maximum_Limit_Value=2047<CR/LF>
```

Channel_Ratio_Primary =2000<CR/LF>
Channel_Ratio_Secondary=1<CR/LF>
Data_Primary_Secondary=P<CR/LF>
<CR/LF>
[Public Analog_Channel_#4] <CR/LF>
Channel_ID=Popular Ia<CR/LF>
Phase_ID=<CR/LF>
Monitored_Component=<CR/LF>
Channel_Units=A<CR/LF>
Channel_Multiplier=11.5093049423<CR/LF>
Channel_Offset=0.0000000000<CR/LF>
Channel_Skew=0<CR/LF>
Range_Minimum_Limit_Value=–2048<CR/LF>
Range_Maximum_Limit_Value=2047<CR/LF>
Channel_Ratio_Primary =1200<CR/LF>
Channel_Ratio_Secondary=5<CR/LF>
Data_Primary_Secondary=P<CR/LF>
<CR/LF>
[Public Analog_Channel_#5] <CR/LF>
Channel_ID=Popular Ib<CR/LF>
Phase_ID=<CR/LF>
Monitored_Component=<CR/LF>
Channel_Units=A<CR/LF>
Channel_Multiplier=11.5093049423<CR/LF>
Channel_Offset=0.0000000000<CR/LF>
Channel_Skew=0<CR/LF>
Range_Minimum_Limit_Value=–2048<CR/LF>
Range_Maximum_Limit_Value=2047<CR/LF>
Channel_Ratio_Primary =1200<CR/LF>
Channel_Ratio_Secondary=5<CR/LF>
Data_Primary_Secondary=P<CR/LF>
<CR/LF>
[Public Analog_Channel_#6] <CR/LF>
Channel_ID=Popular Ic<CR/LF>
Phase_ID=<CR/LF>
Monitored_Component=<CR/LF>
Channel_Units=kV<CR/LF>
Channel_Multiplier=11.5093049423<CR/LF>
Channel_Offset=0.0000000000<CR/LF>
Channel_Skew=0<CR/LF>
Range_Minimum_Limit_Value=–2048<CR/LF>
Range_Maximum_Limit_Value=2047<CR/LF>
Channel_Ratio_Primary =1200<CR/LF>
Channel_Ratio_Secondary=5<CR/LF>
Data_Primary_Secondary=P<CR/LF>
<CR/LF>
[Public Status_Channel_#1] <CR/LF>
Channel_ID=Va over<CR/LF>
Phase_ID=<CR/LF>
Monitored_Component=<CR/LF>
Normal_State=0<CR/LF>
<CR/LF>
[Public Status_Channel_#2] <CR/LF>
Channel_ID=Vb over<CR/LF>

```
Phase_ID=<CR/LF>
Monitored_Component=<CR/LF>
Normal_State=0<CR/LF>
<CR/LF>
[Public Status_Channel_#3] <CR/LF>
Channel_ID=Vc over<CR/LF>
Phase_ID=<CR/LF>
Monitored_Component=<CR/LF>
Normal_State=0<CR/LF>
<CR/LF>
[Public Status_Channel_#4] <CR/LF>
Channel_ID=Ia over<CR/LF>
Phase_ID=<CR/LF>
Monitored_Component=<CR/LF>
Normal_State=0<CR/LF>
<CR/LF>
[Public Status_Channel_#5] <CR/LF>
Channel_ID=Ib over<CR/LF>
Phase_ID=<CR/LF>
Monitored_Component=<CR/LF>
Normal_State=0<CR/LF>
<CR/LF>
[Public Status_Channel_#6] <CR/LF>
Channel_ID=Ic over<CR/LF>
Phase_ID=<CR/LF>
Monitored_Component=<CR/LF>
Normal_State=0<CR/LF>
<CR/LF>
[Company1 event_rec] <CR/LF>
recorder_type=1<CR/LF>
trig_set=0,0,0,0,6048,6272,0,0,0,0,0,0,0,0,0,0<CR/LF>
ch_type=1,1,1,1,1,1,1,0,0<CR/LF>
<CR/LF>
[Company1 analog_rec_1] <CR/LF>
op_limit=15<CR/LF>
trg_over_val=f<CR/LF>
trg_under_val=f<CR/LF>
trg_roc=f<CR/LF>
inverted=0<CR/LF>
```

Annex D

(informative)

Sample program for sampling frequency conversion

```
C    PROGRAM CONVERT
C    CONVERTS SAMPLES TAKEN AT ONE RATE TO A SECOND
C    RATE
C    USER SUPPLIED FILTER IS IN FOR020.DAT
C    DATA IS IN FOR021.DAT
C    OUTPUT IS IN FOR025.DAT
C
C    NFMAX = THE MAXIMUM LENGTH OF THE FILTER
C
     PARAMETER NFMAX = 3600
C    3600 CORRESPONDS TO ONE CYCLE
C
C    LFAC = THE NUMBER OF TENTHS OF A DEGREE BETWEEN
C    SAMPLES IN INPUT
     PARAMETER LFAC=50
C    FSAMP = THE INPUT SAMPLING FREQUENCY
     PARAMETER FSAMP = 4320
C    NSIZE = THE MAXIMUM LENGTH OF THE INPUT DATA
C    STRING
     PARAMETER NSIZE = 720
     INTEGER*2 DBUF(NSIZE)
     DIMENSION HFIL(NFMAX),ZTD1(NFMAX)
     DATA N0/0/
C
C    GET FILTER RESPONSE
     READ(20,*) NA,NB
     IF(NB.LE.NFMAX) GO TO 6
     WRITE(6,5)
5    FORMAT(3X,'DECIMATION FILTER IS TOO LONG')
     STOP
C
6    NBF=NB/LFAC
     IF(NB.EQ.NBF*LFAC) GO TO 10
     WRITE(6,*) 'FILTER LENGTH INDIVISIBLE BY LFAC'
     STOP
C
10   READ(20,*) (HFIL(JJ),JJ=1,NB)
C
C
C*********************************************
C
C
C
C    WRITE(6,18)
18   FORMAT(1H$,'ENTER TOTAL NUMBER OF SAMPLES TO BE
   $ PROCESSED')
```



```
      READ(6,*)ITIME
C
      READ(21,*) (DBUF(JJ),JJ=1,ITIME)
      IPTR=1
C
30    WRITE(6,35)
35    FORMAT(1H$,'ENTER THE DESIRED PROCESSING RATE')
      READ(6,*)DRATE
      MFAC=IFIX(FSAMP*LFAC/DRATE)
      IF(MFAC*DRATE.EQ.FSAMP*LFAC) GO TO 40
C
      WRITE(6,*)'RATE IS UNACHIEVABLE - TRY AGAIN'
      GO TO 30
C
      WRITE(6,*)'INTERPOLATION FACTOR =',LFAC
      WRITE(6,*)'DECIMATION FACTOR =',MFAC
C***********************************
C
      DO 500 I=1,ITIME
      DT=(I-10/4320)
      X=FLOAT(DBUF(IPTR))
      WRITE(26,*) DT,X
C
C
      DO 120 J=1,NBF-1
      INDX=NBF+1-J
120   ZTD1(INDX)=ZTD1(INDX-1)
      ZTD1(1)=X
C
C
      N0=N0+LFAC
      IF(N0.LT.MFAC) GO TO 500
C
      N0=N0-MFAC
C
      ZOUT=0.
      DO 130 J=1,NBF
      INDX=J*LFAC-N0
130   ZOUT=ZOUT+HFIL(INDX)*ZTD1(J)
      ZOUT=ZOUT/FSAMP
      WRITE(25,*) DT,ZOUT
C
500   CONTINUE
      STOP
      END
C****************************************
      PROGRAM FIR
C****************************************
C     IMPULSE INVARIANT DESIGN FOR SECOND ORDER
C     LOW PASS FILTER WITH REAL POLES AT -S1 AND -S2
C
C     TRANSFER FUNCTION = A*S1*S2/(S+S1)(S+S2)
C
C     SAMPLING RATE OF 216000 AT 60 HZ
```

```
C       180000 AT 50 HZ
C
C    ONE CYCLE DURATION FINITE IMPULSE RESPONSE FILTER
C    OBTAINED BY WRITING THE PARTIAL FRACTION
C    EXPANSION OF THE TRANSFER FUNCTION AND FORMING
C    THE IMPULSE RESPONSE IN THE FORM
C    H(T)=SUM{CI*EXP(-SI*T)}
C****************************************
C
     DIMENSION H(3600)
     S1=394.
     S2=2620.
C    MAKE GAIN AT 60 HZ = 1
C    G60=INVERSE OF THE 60 HZ GAIN
C
     G60=(SQRT((S1**2+(377)**2)*(S2**2+(377)**2)))/(S1*S2)
     C1=G60*S1*S2/(-S1+S2)
     C2=G60*S1*S2/(S1-S2)
     WRITE(20,*)1,3600
C
     DO 100 I=1,3600
     DT=(I-1)/216000
     H(I)=C1*EXP(-DT*S1)+C2*EXP(-DT*S2)
     WRITE(20,*)H(I)
100  CONTINUE
     STOP
     END
```

Annex E

(informative)

Example application of conversion factors

This example includes consideration of channel conversion factors (ax + b), primary and secondary ratio factors, and primary/secondary data factor (PS).

Assumptions about the source and form of data follow:

a) Assume a series of sample values representing the values on the primary side of a voltage transformer with a nominal range of ± 40 kV peak supplied through a potential transformer ratio of 400:1.

b) Assume the data is to be stored to represent the primary values.

c) Assume a sampling system resolution of 12 bits, then, in order to preserve accuracy, it is necessary to select a maximum/minimum range greater than the 4096 (± 2048) range of the sampling system.

d) Assume, for simplicity, the decision to simply read the numbers from the device and build all conversion factors in the .CFG file conversion factors "ax + b," but that the data from the recording device represents the value zero as the number 3000, meaning that the data will have a maximum possible value of 5048 and a minimum value of 952.

e) Assume full scale for the sampling device is 120 V secondary.

f) The legal data range for ASCII files is defined in subclause 6.3 is –99999 to 99999, a range of approximately 200000. For binary data files the range is 32767 to –32767, a range of approximately 65000.

The data are to be stored in primary units, therefore:

a) The "PS" variable in the .CFG file should be set to "P" ;
b) The "primary" variable in the .CFG file should be set to 400; and
c) The "secondary" variable in the .CFG file should be set to 1.

The conversion factor "a" is found from the following procedure.

a) Data maximum is x = 5048; data minimum is x = 952.

b) Data range maximum/minimum for sampling device is 4096.

c) Data maximum/minimum occur at ±120 V secondary, or ±120 × 400 (ratio) primary = ±48 000.

d) Primary voltage sample range is ±48 000 = 96 000.

e) Conversion factor "a" is primary voltage sample range/data range:

$$\text{“a”} = 96\,000/4096 = 23.4375 \tag{E.1}$$

The conversion factor "b" is found from the following procedure.

a) Conversion factor "b" is the value that must be added to intermediate value "a" × data (x) to get back to original sample value.
b) Data (x) representing primary voltage of 0 = 3000.
c) Conversion factor "a" = 23.4375 from (E.1)
d) Intermediate value "ax" of data value 3000 = 3000 × 23.4375 = 70312.5.

$$\text{“ax”} = 70312.5, \text{ and “ax + b”} = 0, \text{ therefore “b”} = (0 - 70312.5) = -70312.5 \tag{E.2}$$

Check conversion:

a) Sample maximum = 48 000 V
b) Data maximum (x) = 5048
c) “a” and “b” are from (E.1) and (E.2)
d) Sample = “ax + b” = (23.4375 × 5048) + (–70312.5) = (118312.5) + (–70312.5) = 48000

Annex F

(informative)

Bibliography

[B1] ANSI X3.4-1986 (R1997), Information Systems Coded Character Set—7-Bit American National Standard Code for Information Interchange (7-bit ASCII).[2]

[B2] "Digital Protection Techniques and Substation Function," Final report, CIGRÉ Working Group 34.01, A.G. Phadke (Convener), 1989, CIGRÉ Ref.:38

[B3] Duncan, R., *Advanced MS DOS*, Microsoft Press, 1986.

[B4] IEEE Std 260.1-1993, American National Standard Letter Symbols For Units of Measurement (SI Units, Customary Inch-Pound Units).

[B5] IEEE Std 280-1985 (R1996), IEEE Standard Letter Symbols for Quantities Used in Electrical Science and Electrical Engineering (DoD).

[B6] IEEE Std C57.13-1993, IEEE Standard Requirements for Instrument Transformers.

[B7] Kreyszig, E., *Advanced Engineering Mathematics*, 6th ed. New York: John Wiley & Sons, 1988.

[B8] *MS-DOS Version 6 Operating System User's Guide*, Microsoft Corporation, 1993.

[B9] *Microsoft Windows ™ Version 3.1 Operating System User's Guide*, Microsoft Corporation, 1994.

[B10] *Programs for Digital Signal Processing*, Edited by DSP Committee of the IEEE ASSP Society, 0-87942-128-2, IEEE Press, 1979.[3]

[2]ANSI publications are available from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA (http://www.ansi.org/).

[3]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (http://www.standards.ieee.org/).