IEEE STANDARDS ASSOCIATION

IEEE Recommended Practice for Power Quality Data Interchange Format (PQDIF)

IEEE Power and Energy Society

Sponsored by the
Transmission & Distribution Committee



IEEE
3 Park Avenue
New York, NY 10016-5997
USA

IEEE Std 1159.3™-2019
(Revision of IEEE Std 1159.3-2003)

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1159.3-2019

IEEE Std 1159.3™-2019
(Revision of IEEE Std 1159.3-2003)

IEEE Recommended Practice for Power Quality Data Interchange Format (PQDIF)

Sponsor

Transmission & Distribution Committee
of the
IEEE Power and Energy Society

Approved 8 February 2019

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1159.3-2019

This is a Redline Document produced by Techstreet, a business of Clarivate Analytics

This document is intended to provide users with an indication of changes from one edition to the next. It includes a full-text version of the new document, plus an indication of changes from the previous version.

Redlines are designed to save time and improve efficiencies by using the latest software technology to find and highlight document changes. More professionals are using valuable new technologies like redlines, to help improve outcomes in a fastpaced global business world.

Because it may not be technically possible to capture all changes accurately, it is recommended that users consult previous editions as appropriate. In all cases, only the current base version of this publication is to be considered the official document.

Documents Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1159.3-2019

Redline Processing Notes:

1. ~~**Red Text**~~ - Red strikethrough text denotes deletions.

2. **Blue Text** - Blue underlined text denotes modifications and additions.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1159.3-2019

Abstract: A file format suitable for exchanging power quality related measurement and simulation data in a vendor independent manner is defined in this recommended practice. The format is designed to represent all power quality phenomena identified in IEEE Std 1159-2009, IEEE Recommended Practice on Monitoring Electric Power Quality, other power related measurement data, and is extensible to other data types as well. The recommended file format utilizes a highly compressed storage scheme to help reduce disk space and transmission times. The utilization of Globally Unique Identifiers (GUID) to represent each element in the file permits the format to be extensible without the need for a central registration authority.

Keywords: data interchange, file format, IEEE 1159.3, measurement, monitoring, power quality, PQDIF

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

Copyright © 2019 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1 May 2019. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 978-1-5044-5562-6 STD23557
Print: ISBN 978-1-5044-5563-3 STDPD23557

IEEE prohibits discrimination, harassment, and bullying.
For more information, visit http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html.
No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1159.3-2019

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page, appear in all standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents." They can also be obtained on request from IEEE or viewed at http://standards.ieee.org/IPR/disclaimers.html.

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents (standards, recommended practices, and guides), both full-use and trial-use, are developed within IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association ("IEEE-SA") Standards Board. IEEE ("the Institute") develops its standards through a consensus development process, approved by the American National Standards Institute ("ANSI"), which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE Standards are documents developed through scientific, academic, and industry-based technical working groups. Volunteers in IEEE working groups are not necessarily members of the Institute and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE Standards do not guarantee or ensure safety, security, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims all warranties (express, implied and statutory) not included in this or any other document relating to the standard, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; and quality, accuracy, effectiveness, currency, or completeness of material. In addition, IEEE disclaims any and all conditions relating to: results; and workmanlike effort. IEEE standards documents are supplied "AS IS" and "WITH ALL FAULTS."

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.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1159.3-2019

Translations

The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered or inferred to be the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive 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 comments or questions except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in revisions to an IEEE standard is welcome to join the relevant IEEE working group.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854 USA

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, IEEE does not waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate fee, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance

Copyright © 2019 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. A current IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, 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.

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit IEEE Xplore at http://ieeexplore.ieee.org/ or contact IEEE at the address listed previously. For more information about the IEEE-SA or IEEE's standards development process, visit the IEEE-SA Website at http://standards.ieee.org.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE-SA Website at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

Patents

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 by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at http://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1159.3-2019

Participants

At the time this recommended practice was completed, the P1159.3 Task Force had the following membership:

Daniel Sabin, *Chair*
Joseph Grappé, *Secretary*

Richard Bingham
Ivan Bilić
Math Bollen
Thomas Cooke
Gary Chang
William Dabbs
Steven Johnson
William Howe
Theo Laughner
Kevin Kittredge
Alex McEachern
Jan Meyer
David Mueller
Mathew Norwalk
Marty Page
Scott Peele
Kenneth Sedziol
Michael Schwenke
Steve Tatum
Timothy Unruh
Wilsun Xu
Francisc Zavoda
David Zech

The following members of the individual balloting committee voted on this recommended practice. Balloters may have voted for approval, disapproval, or abstention.

Ali Al Awazi
Richard Bingham
Joe Boyles
Gustavo Brunello
Demetrio Bucaneg, Jr
Mario Manana Canteli
Wen-Kung Chang
William Dabbs
Neal Dowling
Jorge Fernandez Daher
Kamal Garg
Mietek Glinkowski
Joseph Grappe
Thomas Grebé
Randall Groves
Werner Hoelzl
Ronald Hotchkiss
Steven Johnston
Laszlo Kadar
Innocent Kamwa
Peter Kelly
Tanuj Khandelwal
Yuri Khersonsky
Kevin Kittredge
Jim Kulchisky
John McDaniel
Bruce Muschlitz
Matthew Norwalk
Gary Nuzzi
Gearold O. H. Eidhin
Lorraine Padden
Bansi Patel
Craig Preuss
Lakshman Raut
Daniel Sabin
Sergio Santos
Bartien Sayogo
Kenneth Sedziol
Jerry Smith
Gary Smullin
Gary Stoedter
K. Stump
Steve Tatum
David Tepen
Timothy Unruh
James Van De Ligt
John Vergis
Francisc Zavoda
David Zech

Copyright © 2019 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

When the IEEE-SA Standards Board approved this recommended practice on 8 February 2019, it had the following membership:

Gary Hoffman, *Chair*
Vacant Position, *Vice Chair*
Jean-Philippe Faure, *Past Chair*
Konstantinos Karachalios, *Secretary*

Masayuki Ariyoshi
Ted Burse
Stephen D. Dukes
J. Travis Griffith
Guido Hiertz
Christel Hunter
Thomas Koshy
Joseph L. Koepfinger*
Thomas Koshy
John D. Kulick
David J. Law
Joseph Levy
Howard Li
Xiaohui Liu
Kevin Lu
Daleep Mohla
Andrew Myles
Annette D. Reilly
Dorothy Stanley
Sha Wei
Phil Wennblom
Philip Winston
Howard Wolfman
Feng Wu
Jingyi Zhou

*Member Emeritus

Copyright © 2019 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

Introduction

This introduction is not part of IEEE Std 1159.3-2019, IEEE Recommended Practice for Power Quality Data Interchange Format (PQDIF).

IEEE Std 1159.3-2003 provided the power quality industry with the specification for a Power Quality Data Interchange Format (PQDIF) for the transfer of power quality data between power quality monitoring instruments and computers. This transfer standard allows the processing and analysis of power quality measurements using multi-vendor and multi-device measurements. IEEE Std 1159.3 was reaffirmed in 2009.

The IEEE P1159.3 Working Group completed the following changes in this revision of IEEE Std 1159.3:

- Completed editorial changes and corrections to the 2003 edition of IEEE Std 1159.3
- Added new ID values for existing PQDIF tags in normative Annex B.
- Added new tags and ID values in normative Annex B.
- Added new quantity types in normative Annex B.
- Added an informative Annex C.2, which specifies an XML directory for PQDIF files that can be downloaded from a remote power quality monitor or remote file server.
- Added an informative Annex D on the representation of PQDIF in XML.
- Added an informative Annex E on PQDIF and its relationship to IEEE Std C37.111 COMTRADE.
- Added an informative Annex F on PQDIF and its relationship to IEC 61850.

The physical structure of PQDIF remains unchanged between IEEE Std 1159.3-2003 and this new edition, making PQDIF both backward and forwards compatible. Changes to the logical structure to support new capability are in the form of additional tag and ID values. This allows software that reads PQDIF to maintain compatibility by simply ignoring unrecognized tag and ID values.

The primary user of this standard will be engineers and software developers who are called upon to create software to encode and decode PQDIF files utilized for power quality data interchange using the guidelines in this recommended practice. Although the document is thorough and complete, detailed examples and source code can be quite helpful when implementing the format described herein. Therefore, reference and supporting documentation (header files, sample code, etc.) related to this standard can be found at the IEEE 1159.3 task force website, which at the time of this writing could be found at the following URL: http://grouper.ieee.org/groups/1159/3/docs.html

More published information about PQDIF is available. An overview of IEEE Std 1159.3 is available in Gunther [B2].[1] An example of how PQDIF was used as part of electric utility power quality monitoring program is described in Veloso, Lebre, and Melo [B18]. An overview of the IEEE 1159.3 revision project is available in Sabin and Dabbs [B17].

[1]The numbers in brackets correspond to those of the bibliography in Annex G.

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1159.3-2019

Contents

1. Overview 10
1.1 Scope 10
1.2 Purpose 10

2. Normative references 10

3. Definitions 11

4. Physical structure 11
4.1 Overall file structure 11
4.2 Internal record structure 14

5. Logical structure 15
5.1 Logical record structure 15
5.2 Logical element structure 16
5.3 Logical structure of container record 16
5.4 Logical structure of data sources and observations 18
5.5 How to assemble a series 19

6. Compression 21
6.1 Compression algorithm—zlib 21
6.2 Record-level compression 22

7. Logical device modeling 22
7.1 General procedures 23
7.2 Representing a waveform recording 26
7.3 Representing a RMS variation recording 29
7.4 Representing periodically recorded steady state values 36
7.5 Representing magnitude-duration-time tables 41

Annex A (normative) Physical format definitions 44
A.1 Physical format definitions—pqdif_ph.h 44

Annex B (normative) Logical structure documentation 52
B.1 Logical format constant definitions—pqdif_lg.h 76
B.2 Constant IDs and GUIDS 99

Annex C (informative) PQDIF file storage and file download 159
C.1 File storage and management 159
C.2 PQDIF file server interfaces 162

Annex D (informative) Representation of PQDIF in XML 168
D.1 Examples 170

Annex E (informative) Relationship of PQDIF to COMTRADE 176
E.1 PQDIF to COMTRADE 176

Annex F (informative) Relationship of PQDIF to IEC 61850 181
F.1 IEC TR 61850-90-17 181

Annex G (informative) Bibliography 182

Copyright © 2019 IEEE. All rights reserved.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1159.3-2019

IEEE Recommended Practice for ~~the Transfer of~~ Power Quality Data Interchange Format (PQDIF)

1. Overview

1.1 Scope

This recommended practice specifies the PQDIF file format for the transfer of power quality data between instruments and computers. This includes raw, processed, simulated, proposed, specified, and calculated data. The transfer file format includes the power quality measurements as well as appropriate characterization parameters, such as sampling rate, resolution, calibration status, instrument identification, location, and other related data or characteristics. The recommended practice also provides guidelines for transferring power quality data.

~~Develop a recommended practice for a file format suitable for exchanging power quality related measurement and simulation data in a vendor independent manner. Appropriate definitions and event categories to be developed by other task forces under the IEEE Standards Coordinating Committee 22 (SCC 22) on Power Quality and the IEEE 1159 Working Group on Power Quality Monitoring.~~

1.2 Purpose

IEEE Std 1159.3 provides the power quality industry with the specification for PQDIF, which is an open and accepted data format standard for the transfer of power quality data between instruments and computers. This transfer standard allows the processing and analysis of power quality measurements using multi-vendor and multi-device data. Wider acceptance of PQDIF as a power quality data transfer format will significantly add to the value of power quality monitoring and open new opportunities for the resolution, planning, and understanding of power quality activities.

Being able to exchange data between software systems will allow other functions needed in a power quality monitoring campaign, including validation, trending, comparison, overlay, and more.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

A variety of simulation, measurement, and analysis tools for power quality engineers are now available from many vendors. Generally, the data created, measured, and analyzed by these tools are incompatible between vendors. The proposed file format will provide a common ground that all vendors could export to and import from to allow the end user maximum flexibility in choice of tool and vendor.

There are two "layers" to this Power Quality Data Interchange Format (PQDIF): the physical layer and the logical layer. The physical layer describes the physical structure of the file without regard to what will actually be stored in it. It uses tags to identify particular elements of the file. This is similar in concept to tagged image file format (TIFF), used for storing images.

The logical layer uses the structure defined by the physical layer; it specifies specific tags to use when building up elements in the file.

The physical layer is based on

- Specific "physical" data types (e.g., INT1, INT2, INT4, REAL4, REAL8, etc.) for portability and a specific list of IDs for physical representation (e.g., ID_SERIES_PHYS_TYPE_INTEGER1, etc.)
- 4 byte alignment for efficient processing
- Little endian byte ordering (least significant byte stored first, ordering used in PCs)
- Tags, using globally unique identifiers (GUIDs), for unique identification of elements (hereafter called "tags")

The logical layer is based on

- Specific lists of tags to identify elements of a file
- A hierarchy of tags and expected physical types
- Extensibility using user-defined tags for private data
- Extensibility of the standard format using tags defined in the future

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

~~To keep things simple, many elements in the logical layer are based on an explicit list of enumerated IDs, such as~~

- ~~Phase (ID_PHASE_AN, ID_PHASE_BN, etc.)~~
- ~~IEEE 1159 disturbance category (ID_1159_TRANSIENT, ID_1159_SHORTDUR, etc.)~~
- ~~High-level quantity type (ID_QT_WAVEFORM, ID_QT_RMS, etc.)~~
- ~~Series quantity units (ID_QU_TIMESTAMP, ID_QU_VOLTS, ID_QU_AMPS, etc.)~~
- ~~Series value type (ID_SERIES_VALUE_TYPE_MIN, ID_SERIES_VALUE_TYPE_MAX, etc.)~~

~~Once you understand the physical and logical formats of PQDIF, you will still need some guidelines on how to use it to represent real-world data.Please refer to Annex A and Annex B for details on how to represent various types of data. Also, third parties have created PQDIF tools and development kits to facilitate getting up to speed on PQDIF. The IEEE 1159 working group web site (http://grouper.ieee.org/groups/1159/3/doc.html) has public domain example data files, programs, and source code. Annex C documents a commonly implemented disk directory structure for containing multiple PQDIF files.~~

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEC 61000-4-30, Electromagnetic compatibility (EMC)—Part 4-30: Testing and measurement techniques—Power quality measurement methods.[1]

[1] IEC publications are available from the International Electrotechnical Commission (http://www.iec.ch) and the American National Standards Institute (http://www.ansi.org/).

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

~~This recommended practice shall be used in conjunction with the following publications. When the following publications are superseded by an approved revision, the revision shall apply.~~

IEEE Std 1159™~~-1995 (Reaff 2001)~~, IEEE Recommended Practice on ~~for~~ Monitoring Electric Power Quality.[2, 3] ~~[1]~~

IETF RFC 1950, ZLIB Compressed Data Format Specification~~,~~ version 3.3.[4] ~~[2]~~

IETF RFC 1951, DEFLATE Compressed Data Format Specification~~,~~ version 1.3.

3. Definitions

For the purposes of this document ~~recommended practice~~, the following terms and definitions apply. The ~~*Authoritative Dictionary of*~~ *IEEE Standards*

Dictionary Online ~~*Terms*~~ should be consulted ~~referenced~~ for terms not defined in this clause.[5]

~~3.1 globally unique identifier (~~GUID~~):~~ An acronym for Globally Unique Identifier, which is a randomly-generated 16-byte value used in PQDIF to uniquely identify an object, which ~~that when randomly generated~~ is large enough to be considered globally unique for all practical purposes ~~(infinitesimal chance of another party randomly generating the same value). It is used to uniquely identify an object~~.

[2] The IEEE standards or products referred to in Clause 2 are trademarks owned by the Institute of Electrical and Electronics Engineers, Incorporated.

[3] ~~[1]~~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://standards.ieee.org/).

~~[2]RFC documents are available from Requests for Comments Editor, http://www.rfc-editor.org/.~~

[4] IETF RFCs are available at https://www.ietf.org/standards/rfcs/.

[5] *IEEE Standards Dictionary Online* subscription is available at: http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

PQDIF: An acronym for Power Quality Data Interchange Format, which is a file- or stream-based mechanism of transferring power quality and related measurement or simulation data from a data source to a data sink in a compact, unambiguous, and deterministic way.

PQDIF file: A collection of PQDIF records consisting of a Container record, Data Source record, an optional Monitor Settings record, and one or more Observation records.

PQDIF stream: A collection of PQDIF records streamed to a client using communication media such as a network connection. A stream presents records such that they are downloaded in the same order as they would be if read from a PQDIF file.

4. Physical structure

PQDIF achieves its flexibility by segregating the definition of its physical structure from that of its logical structure. The physical structure defines how basic and complex data types are constructed in a file; the logical structure defines how the physical data is interpreted as power quality data.

4.1 Overall file structure

4.1.1 Standard PQDIF flat file

The file is made up of a series of records that are arranged in a linked list. These links are provided by an *absolute link* (absolute file offset or position) in the header of the records. This allows new records to be inserted, and old records to be deleted. For the records shown in Figure 1, the order of the records is Record 1, Record 2, Record 3 … Record *n*.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

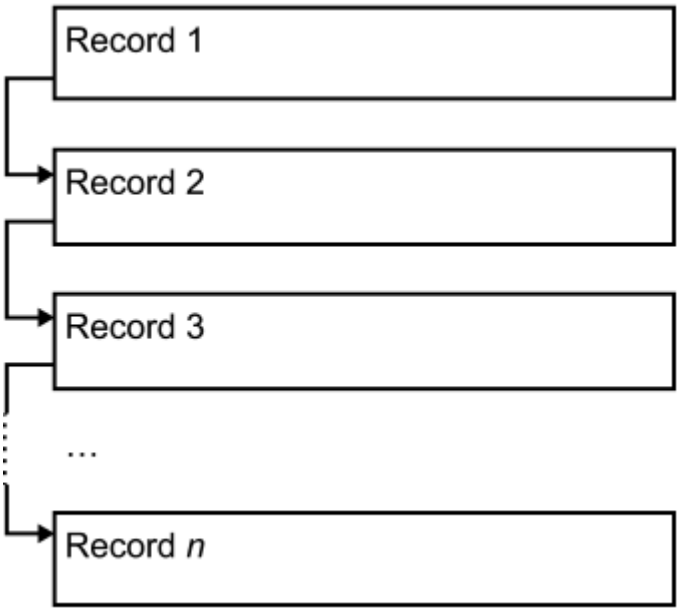


Figure 1—Standard record linkages

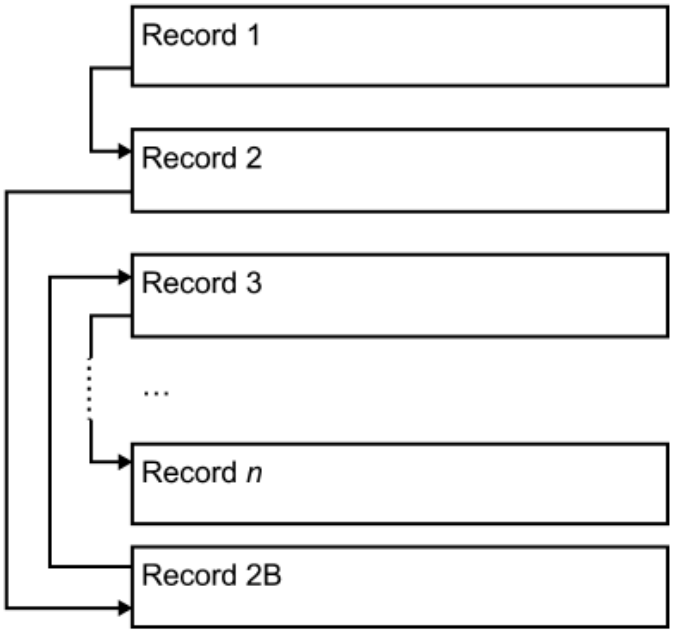


Figure 2—Inserting a new record

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

After the operation in Figure 2, the order of the records is now Record 1, Record 2, Record 2B, Record 3, … Record *n*.

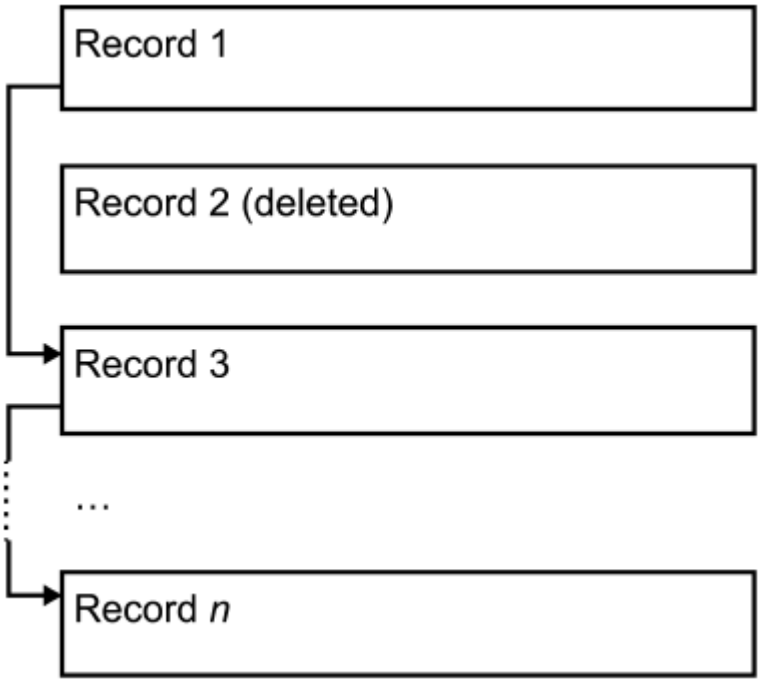


Figure 3—Deleting a record

After the operation in Figure 3, the order of the records is now Record 1, Record 3 … Record *n*.

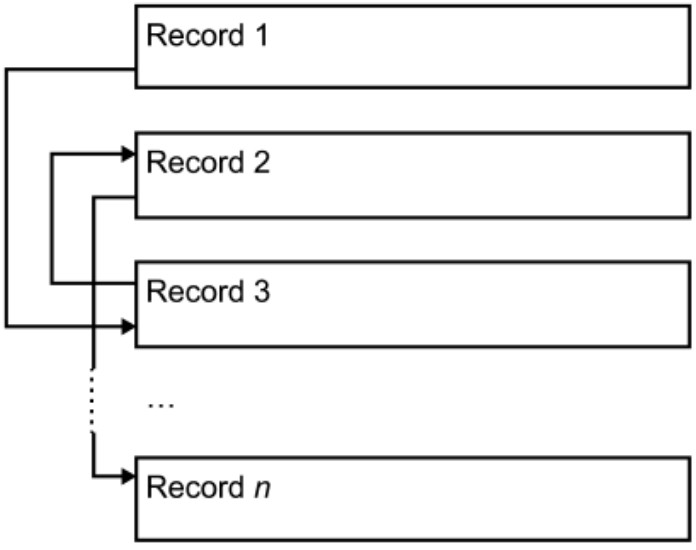


Figure 4—Reordering records

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

After the operation in Figure 4, the order of the records is now Record 1, Record 3, Record 2 … Record *n*.

4.1.2 Alternate PQDIF structures

The PQDIF format is flexible enough that other structures besides the flat file can be used. For example, it would be possible to store individual PQDIF records in a database or a structured storage medium. ~~of some sort.~~ In this case, some of the items in the header (the *absolute link*, for example) are not necessary and can be ignored. In some cases, the header can be discarded entirely. Using PQDIF in this fashion is out of the scope of this document, since ~~because~~ the primary focus is the standard flat file that will be used for interchanging data.

See Annex D for an example of storing PQDIF records in XML format.

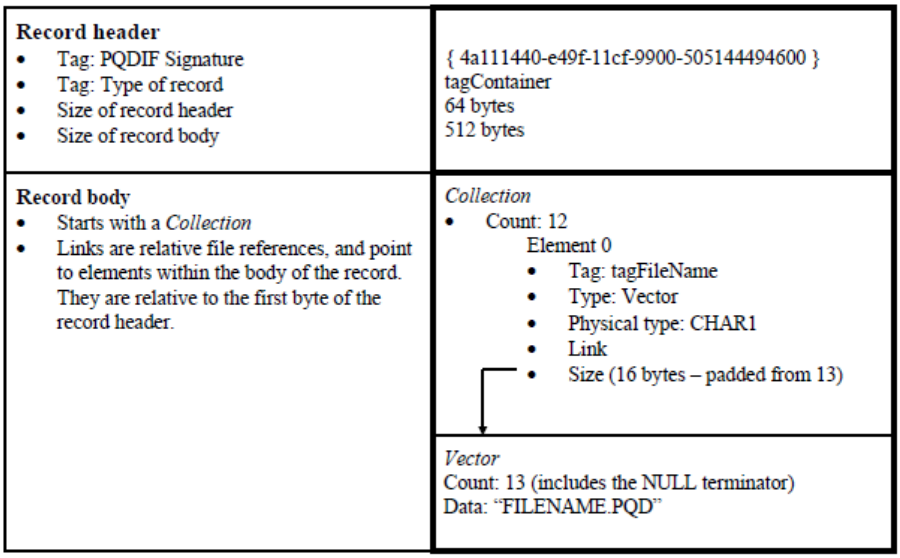
Some implementations of PQDIF involve use of a PQDIF Stream, in which a collection of PQDIF records are read by a client using a communications connection. The records in a stream are provided to a client in the same order that they would appear if read from a PQDIF file. The advantage to this approach is that the measurements could be used by a PQDIF reader before the complete PQDIF file is downloaded.

4.2 Internal record structure

Each record type in a PQDIF file has the same basic structure consisting of a header and then the record body. Table 1 illustrates the structure of a record.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

Table 1 —Record structure



4.2.1 Record header

Each record has a standard header, which is marked with a unique "signature"—in this case, a Globally Unique Identifier (GUID ~~shown below~~). The other items in the header specify the *tag* of the record (container, data source, etc.), its *size*, and the *absolute link* to the next record. There may be others, such as *status* (active/deleted).

4.2.2 Record body

The record body is made up of a set of *elements*. There are three types of elements: *Collection*, *Scalar*, and

Vector.

— A collection is essentially an array of *tags* and *relative links* to other elements. Because a collection can link to another collection, it is simple to create a hierarchical structure.

— A scalar is a single value of a specific physical type (INT4, REAL8, etc.).

— A vector is an arbitrarily-sized array ~~array~~ of a specific physical type.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

Each element is given a *tag* by the collection that contains it (except for the first collection). The record body always starts with a collection element. This element is tagged by its location in the record—~~,~~ i.e., the first element. Therefore, it is identified by the tag found in the record header.

A *relative link* is a file position that is relative to the beginning of the record header.

5. Logical structure

5.1 Logical record structure

The basic structure is made up of a logical hierarchy of "records"—consisting of a single container, followed by one or more data sources, monitor settings, and observations. There can also be "extended" records by defining new tags.

The order of the records is defined by the absolute links in the header of each record. By following the list of records, a linear list of records is defined. This linear list has a logical hierarchy defined as shown in Figure 5.

Note that several monitor settings records can appear in the middle of the observations for a specific ~~particular~~ data source. This allows the monitor (instrument) setup to change. Consequently, some parameters in the observations may depend on the information in the appropriate monitor settings record. If ~~we did not allow~~ multiple settings records were not allowed, a totally new data source would have to be created if any settings changed.

The Date Effective time stamps of data source and monitor settings records are used to determine which of these records applies to a given observation record. The data source and monitor settings records that have ~~with~~ a Date Effective time stamp equal to or earlier than ~~prior to~~ the subject observation record should be ~~are~~ used.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

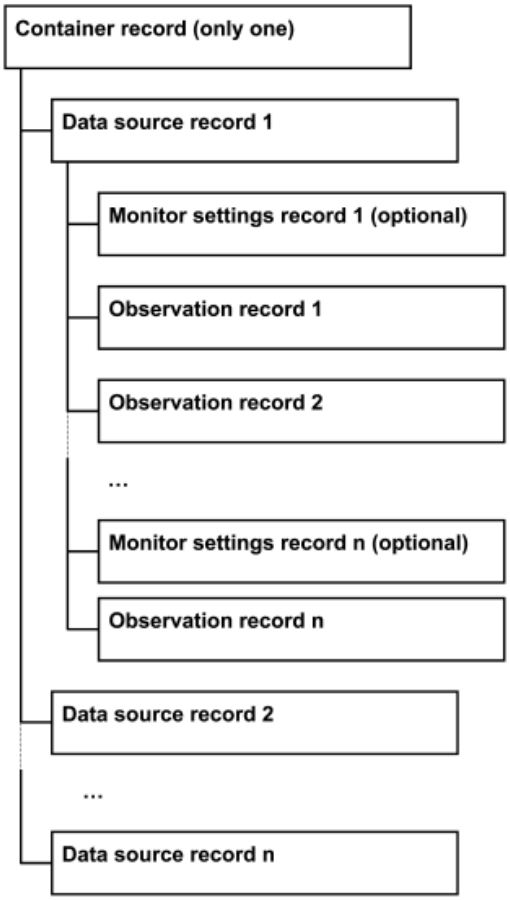


Figure 5—Logical record structure

5.2 Logical element structure

5.2.1 Introduction

The complete logical format is defined by a set of tags, and an expected structure associated with each tag. For example, a *tagName* element must, by definition, be a vector of CHAR1—~~,~~ in other words, a string. It has no expected length (i.e., it can be as long or as short as necessary). A *tagVersionInfo* element is ~~slightly~~ more complex: it must be a vector of UINT4~~,~~ and ~~it~~ must have four entries~~:~~ major and minor version, and major and minor version "compatible."

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

Even more complex is the way hierarchies are defined. A *tagChannelDefns* element must be a collection, and ~~it~~ must contain another set of collections, which are all tagged with *tagOneChannelDefn*. (The number of elements in the *tagChannelDefns* element defines how many channel definitions are available.) Inside each *tagOneChannelDefn* collection is another set of required tags, and so forth.

See Annex A and Annex B for full documentation of all tags. See the C language header files (Table 2) included in these documents for the actual tag definitions and for the lists of IDs.

Table 2 —C Language implementation header files

Header file	Contents
PQDIF_PH.H	All data structure declarations for the physical structure (Annex A).
PQDIF_LG.H	All tag declarations for the physical structure. (These tags are documented in Annex B.)
PQDIF_ID.H	An extensible list of ID declarations for things such as vendors and equipment. (These IDs are documented in Annex B.)

5.3 Logical structure of container record

The container record contains overall, ~~,~~ or summary, ~~,~~ information about the PQDIF file. The following text shows an ASCII dump of the container record from a typical PQDIF file produced by a simple PQDIF utility program.

```
-Collection -- tag: tagContainer (level 0)
 +-Vector -- tag: tagFileName (type: CHAR1) [ 32 ]
 | value: 'D:\PQDIF\Native15\PQDIFr\wf.pqd'
 +-Scalar -- tag: tagCreation (type: TIMESTAMPPQDIF)
 | value: 7/7/1999 20:40:58.000000217
 +-Vector -- tag: tagVersionInfo (type: UNS_INTEGER4) [ 4 ]
 | values: 1, 5, 1, 5
 +-Vector -- tag: tagLanguage (type: CHAR1) [ 11 ]
 | value: 'US English'
 +-Vector -- tag: tagTitle (type: CHAR1) [ 25 ]
 | value: 'PQDIF Translator'
 +-Vector -- tag: tagSubject (type: CHAR1) [ 15 ]
 | value: 'PQDIF Export'
 +-Vector -- tag: tagAuthor (type: CHAR1) [ 17 ]
 | value: Thomas Edison'
 +-Vector -- tag: tagKeywords (type: CHAR1) [ 37 ]
 | value: 'PQDIF Translator for IEEE'
 +-Vector -- tag: tagComments (type: CHAR1) [ 24 ]
 | value: 'Comments field not used'
 +-Vector -- tag: tagLastSavedBy (type: CHAR1) [ 27 ]
 | value: 'lastSavedBy field not used'
 +-Vector -- tag: tagApplication (type: CHAR1) [ 34 ]
 | value: 'PQDIF translator for IEEE'
 +-Vector -- tag: tagSecurity (type: CHAR1) [ 12 ]
```

(New)

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

```
-Collection -- tag: tagContainer (level 0)
   +-Vector -- tag: tagFileName (type: CHAR1) [ 32 ]
   |  value: 'D:\PQDIF\Native15\PQDIFr\wf.pqd'
   +-Scalar -- tag: tagCreation (type: TIMESTAMPPQDIF)
   |  value: 7/7/1999 20:40:58.000000217
   +-Vector -- tag: tagVersionInfo (type: UNS_INTEGER4) [ 4 ]
   |  values: 1, 5, 1, 5
   +-Vector -- tag: tagLanguage (type: CHAR1) [ 11 ]
   |  value: 'US English'
   +-Vector -- tag: tagTitle (type: CHAR1) [ 25 ]
   |  value: 'PASS to PQDIF Translator'
   +-Vector -- tag: tagSubject (type: CHAR1) [ 15 ]
   |  value: '8010/20 Export'
   +-Vector -- tag: tagAuthor (type: CHAR1) [ 17 ]
   |  value: 'Erich W. Gunther'
   +-Vector -- tag: tagKeywords (type: CHAR1) [ 37 ]
   |  value: 'PASS 8010 8020 PQDIF Translator IEEE'
   +-Vector -- tag: tagComments (type: CHAR1) [ 24 ]
   |  value: 'Comments field not used'
```

~~(Deleted)~~

```
| value: 'No security'
+-Vector -- tag: tagOwner (type: CHAR1) [ 26 ]
| value: ABCDE Corporation
+-Vector -- tag: tagCopyright (type: CHAR1) [ 67 ]
| value: 'Copyright (C) 2017 ABCDE Corporation, Inc. All Rights Reserved.'
+-Vector -- tag: tagTrademarks (type: CHAR1) [ 75 ]
| value: ABCDE Corporation is a registered trademark of ABCDE Corporation, Inc.'
+-Vector -- tag: tagNotes (type: CHAR1) [ 9 ]
| value: 'No notes'
+-Scalar -- tag: tagCompressionStyleID (type: UNS_INTEGER4)
| value: 2
+-Scalar -- tag: tagCompressionAlgorithmID (type: UNS_INTEGER4)
| value: 1
+-(End of collection)
+---- Record information
| Size on disk: 1028
+-(End of record)
```

(New)

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1159.3-2019



~~(Deleted)~~

The following ~~There are~~ two pieces of information ~~that~~ are particularly important when interpreting the contents of the file:

— Version

— Compression

5.3.1 Version

Version information is used to control and track the evolution of the format. At some point, major changes may need to be made that affect the interpretation of some of the tags. If these changes are drastic, older readers may not be able to interpret these standard tags properly. In this case, the "compatible version" (as described below) may have to be bumped up so that older readers will know not to attempt interpretation.

On the other hand, it may be possible to continue to update the standard without affecting backward compatibility. In this case, the compatible version will remain at 1.0, and all older readers will be able to interpret the file without~~, sans~~ any new tags~~, of course~~.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

Newer readers will be able to tell what tags to expect by looking at the "writer version." Basically, all readers should do the following:

— If it can support the writer version, read the file using this version.

— Otherwise, if it can support the compatible version, read the file using this version.

— If it cannot support either, do not attempt to read the file.

This information is contained in a vector tagged with *tagVersionInfo*. This vector has four members, ~~as~~ denoted as follows in Table 3.

Table 3 —Version-related items

Index	Name	Meaning
0	Writer major version	The version of the logical structure being used by the writer of the file. This is for a reader to determine.
1	Writer minor version	"
2	Compatible major version	The file is also compatible with this version of the logical structure. Older readers can use this to determine if they will be able to read this file.
3	Compatible minor version	"

(New)

~~(Deleted)~~

Index	Name	Meaning
0	Writer major version	The version of the logical structure being used by the writer of the file. This is for a reader to determine.
1	Writer minor version	The version of the logical structure being used by the writer of the file. This is for a reader to determine.
2	Compatible major version	The file is also compatible with this version of the logical structure. Older readers can use this to determine if they will be able to read this file.
3	Compatible minor version	The file is also compatible with this version of the logical structure. Older readers can use this to determine if they will be able to read this file.

NOTE—The version information is intended to apply *only* to the logical structure; —the physical structure will remain standard throughout the life of the format.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

The previous ASCII dump ~~above~~ indicates that the subject PQDIF file was written by a program that uses ~~using~~ version 1.5 and it is compatible with version 1.5 and later reader programs.

5.3.2 Compression

The container record contains an element indicating whether compression is used. The ASCII dump above specifies the tagCompressionStyleID and the tagCompressionAlgorithmID. The values shown indicate that the PQDIF file uses record level compression and the zlib compression algorithm is used. The integers used for these types are defined in Annex B. See Clause 6 for~~, which provides~~ more information on compression.

5.4 Logical structure of data sources and observations

5.4.1 Structure

The internal structure of the *data source* and *observation* records parallel each other, and in so doing, ~~they~~ are implicitly linked together (Figure 6). In object-oriented design language, it can be described this way (note that "many" is a technical term here; it ~~actually~~ means 0 or more):

A *data source* **has** many *channel definitions*; each *channel definition* **has** many *series definitions*. An *observation* **has** many *channel instances*; each *channel instance* **has** many *series instances*.

This definition/instance structure parallels the object-oriented class/instance structure, and ~~it~~ serves the same purpose. "Static" information common to all "instances" is kept at the definition (or class) level.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

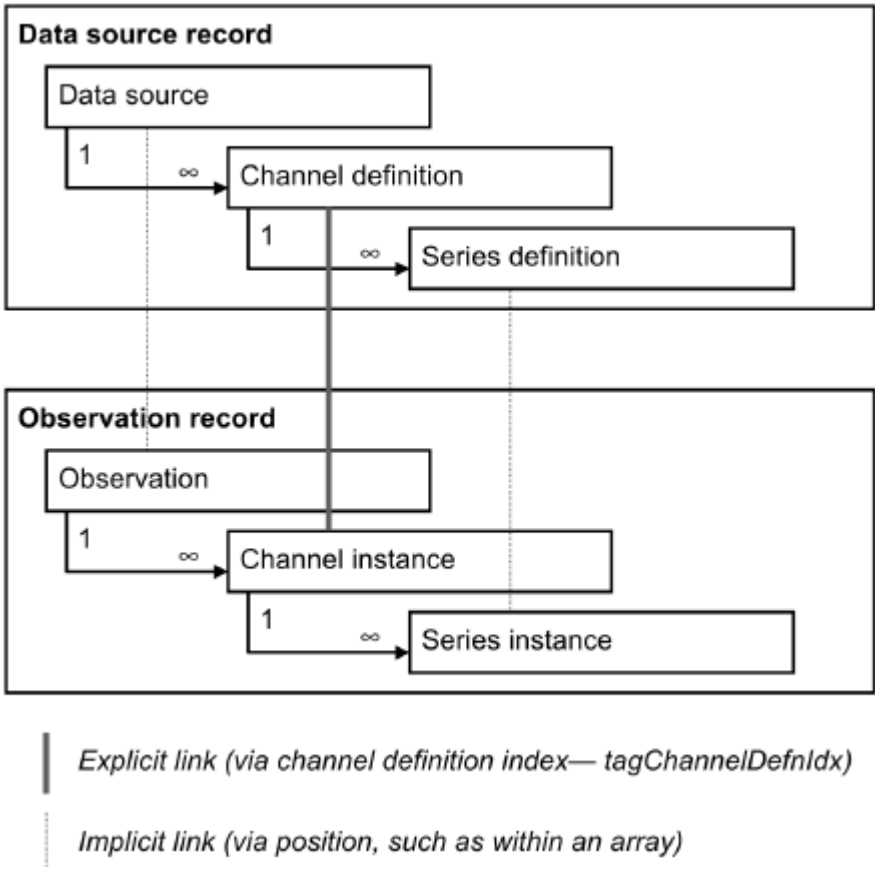


Figure 6—Data Source and observation record relationship

5.4.2 Definitions

The channel definition (*tagChannelDefn*) holds the following information about the channels that would not need to be repeated for each observation~~, as follows~~:

— Channel Name

— Phase or other name

— Name or ~~if~~ phase not applicable

— High-level quantity type (waveform, trend, etc.)

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

— Group name (Bus 1, Bus 2, XMFR ~~XFMR~~ High Side, etc.)

The series definition (*tagSeriesDefn*), likewise, holds the following information that is common to all observations~~, as follows~~:

— Quantity Measured (voltage, current, power, etc.)

— Quantity units (volts, amps, etc.)

— Quantity characteristic (instantaneous, peak, rms, etc.)

— The series value type (Min, Max, Average, etc.)

— Hints about how to display the data (expected Greek prefix, per unit vs. engineering units, etc.)

5.4.3 Instances

The specific channel instance (*tagChannelInstance)* contains ~~actually holds very~~ little data except providing a link to the series instances. However, every channel definition *may or may not* be used by every single observation. Thus, there may be fewer channel instances than there are ~~we have~~ channel definitions. The channel instance then uses a definition index (*tagChannelDefnIdx*) to indicate that it is an instance of a specific channel definition.

In other words, the channel definitions provide a "library" of possible channels to be "instantiated." The individual channel instance can decide which definition to use. It is not necessary for each observation to use every possible channel definition; the *tagChannelDefnIdx* tag is used instead to specify which channel definitions should be ~~are actually~~ used.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

Similarly, there may be more than one instance of a specific ~~particular~~ channel definition. This is most often encountered when reporting multiple attributes of a specific ~~particular~~ measured quantity characteristic. For example, if the third, fifth, and seventh harmonic group magnitudes of the voltage represented by a specific ~~particular~~ definition are ~~is~~ to be recorded, three instances of this channel definition are stored in the file. Each instance contains an element that specifies ~~for~~ which harmonic (more generally the frequency) the instance represents. A similar technique is used to represent probability data. Multiple instances of the same channel are used to record data for a specified percentile.

The series instance (*tagSeriesInstance*) holds the array of data values that makes up its portion of the observation. The series instance is a little different from the channel instance in that each series instance must match up directly with each series definition. If there is a need for different series topologies, ~~you will need to create~~ additional channel definitions will be needed for each one.

5.5 How to assemble ~~put together~~ a series

Proper use of the series options described in this clause ~~section~~ can result in a significant reduction in the size of a PQDIF file. Methods are provided to represent data series as scaled integers of varying size, referenced copies of another series, or series represented by a simple count and interval. Use of these options is recommended to help reduce ~~minimize~~ PQDIF file size.

5.5.1 Explicit and scaled series

Most series will be a straight vector of data values, typically of the types INT1, INT2, INT4, REAL4, or REAL8. The vector's count will be the total number of data points in the series. In this case, the storage method will be ID_SERIES_METHOD_VALUES.

This can be evaluated with a logical OR ~~'d~~ with ID_SERIES_METHOD_SCALED to scale each value by another constant. This would typically be used when the "raw" sample (the 2-byte integer return by an A/D converter, for example) is being stored. This way, the PQDIF file is smaller, and the final scaled value (which might otherwise have to be stored as a REAL4 or REAL8) can be calculated at read time.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

5.5.2 Regular rate series

In some cases, there ~~you~~ may be ~~have~~ a series in which each number is separated by a constant value (this happens often in time series with a constant sampling rate). To represent this, ~~you can put together~~ an array of numbers can be put together that indicates how far apart the values are supposed to be. The storage method will be ID_SERIES_METHOD_INCREMENT. This can even have multiple sampling rates~~,~~ as follows:

```
vector, count = ( N * 2 ) + 1
[ 0 ] = N (total number of rate changes)
[ 1 ] = number of points (0)
[ 2 ] = rate (0)
...
[ N * 2 - 1] = number of points (N-1)
[ N * 2 ] = rate (N-1)
```

Thus, if there is ~~you had~~ a single sampling rate (say, 0.01 seconds) for 1792 points, the series would be as follows:

```
vector, count = 3
[0] = 1
[1] = 1792
[2] = 0.01
```

In addition, ~~you can OR~~ the storage method can be evaluated with a logical OR with ID_SERIES_METHOD_SCALED to represent the sample rate. In this case, the series would be as follows:

vector, count = 3

[0] = 1

[1] = 1792’d

[2] = 1

and the *tagSeriesScale* element would contain 0.01.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

Naturally, if there are ~~you had~~ multiple sampling rates, this scale factor ~~you~~ would probably not be used, ~~use this scale factor~~ and ~~would~~ instead ~~put~~ the scaling factor would be put directly into the series, as shown here. The following ~~This~~ example shows a series that still contains 1792 points, but has two different sampling rates~~, as follows~~:

```
vector, count = 5
[0] = 2
[1] = 896
[2] = 0.01
[3] = 896
[4] = 0.02
```

If the ~~your~~ regular rate series does not start at 0, ~~you can add~~ a *tagSeriesOffset* element can be added as a starting point. (Note that ~~—~~for time series, the first value should always be 0.)

5.5.3 Shared series

For a normal series, a *tagSeriesValues* vector is required. However, ~~in some cases, you may want~~ to save space, sharing ~~share~~ data from another series is an option~~, for space savings~~. In this case, the *tagSeriesValues* tag is replaced by two tags: *tagSeriesShareChannelIdx* and *tagSeriesShareSeriesIdx*. These tags indicate a "master" normal series (e.g., one that has a *tagSeriesValues* tag)~~,~~ which has the designated data ~~we want~~ to be shared ~~share~~.

These tags (*tagSeriesShareChannelIdx* and *tagSeriesShareSeriesIdx*) are indices into collections within the current observation.

5.5.4 Use of IEEE special floating-point numbers

Special floating-point numbers specified in IEEE Std 754 [B8] should not be stored in a PQDIF series instance. Specific examples of special numbers that should not be used include positive infinity, negative infinity, not a number (NaN) values, quiet NaNs, and signaling NaNs.

Missing data should be handled by ending a series instance and beginning a new series instance. Intervals that need to be flagged should be use the flagging procedure described in 5.5.5.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

5.5.5 IEC 61000-4-30 flagging

Some standards specify that values should be flagged to allow data readers to treat values in a special way. IEC 61000-4-30[6] employs the concept of "flagging" certain aggregated measurements. Data is flagged if it was recorded during an interval that includes a voltage sag, a voltage swell, or a voltage interruption.

To store flagging information, the PQDIF file should contain a channel definition record with tagQuantityTypeID equal to ID_QT_VALUELOG and tagQuantityMeasuredID equal to ID_QM_STATUS.

The channel definition record should have two series definition records. The first should have tagValueTypeID equal to ID_SERIES_VALUE_TYPE_TIME. The second should have tagQuantityCharacteristicID set to ID_QC_IEC_61000_4_30_FLAGGING and tagValueTypeID set to ID_PHYS_TYPE_BOOLEAN1, ID_PHYS_TYPE_BOOLEAN2, or ID_PHYS_TYPE_BOOLEAN4.

The intervals that need to be flagged can then be stored as individual observation records with the tagTimeStart equal to the conclusion time of the interval to be flagged, and with a series value set to 1 (for true).

Alternatively, a single observation record could be used with a series of values set to 1 if the ten-minute interval is flagged or 0 if the ten-minute interval is not flagged.

6. Compression

6.1 Compression algorithm—zlib

Since ~~As~~ PQDIF is a binary format, it is ~~fairly~~ compact even when uncompressed. Nevertheless, compression can still help in many cases. PQDIF uses the public domain *zlib* compression library when compression is desired. This algorithm is a ~~an~~ LZ77 variant (like ~~similar to~~ PKZIP) and is not covered by any patents (unlike the LZW algorithm). The zlib home page can be found at http://zlib.net/ ~~http://www.gzip.org/zlib/~~.

[6] For information on references, see Clause 2.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

The data format used by the zlib library is described by IETF RFC 1950 and IETF RFC 1951.

6.2 Record-level compression

In record-level compression, all record headers are uncompressed (Figure 7). All record *bodies* (except the container record) are compressed as individual blocks. This compression method allows a PQDIF reader program to quickly discover the structure of the file without having to read and decode the entire contents of the file.

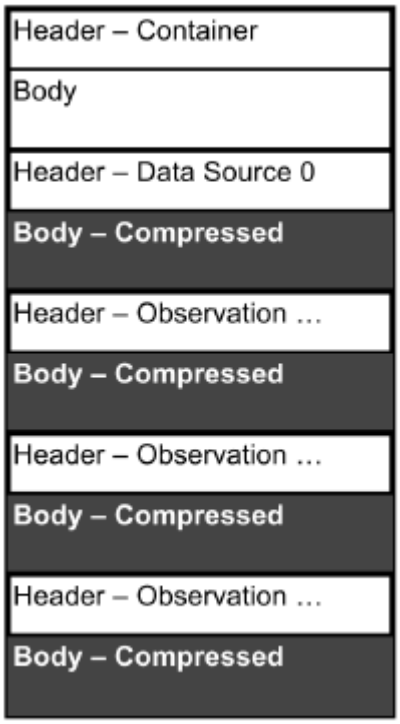


Figure 7—Compressed record layout

Since ~~As~~ the headers are uncompressed, the absolute links work as before. The header (and the links, of course) must reflect the *compressed* size of the body, not the uncompressed size. The relative links within the body work as before—~~:~~They are all set before compression occurs. ~~In other words, create the body and then compress it.~~ A checksum (a 32-bit CRC) for each record body must be stored in the appropriate record header.

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 1159.3-2019

7. Logical device modeling

The previous clauses ~~Clause 4 and Clause 5~~ described the core format itself, introduced the concept of data source, monitor settings, and observation records and how to represent a basic data series. This clause ~~section~~ describes the process of modeling power quality data in more detail. PQDIF is designed to support representing simulated as well as measured power quality data. For the purposes of this clause, the focus ~~section, we~~ will be ~~focus~~ on power quality monitoring instrument modeling.

~~The first thing we must realize is that~~ PQDIF is not intended to be an optimal format for use as a native information store for an actual instrument. For some instruments it could be, but not generally. More often, an instrument vendor will choose to store information recorded in their ~~its~~ instrument in an ~~a manner~~ efficient manner for the internal architecture of the device. A PQDIF file is only created when that information needs to be extracted in an instrument-independent manner for some undefined, external use. This translation may occur in the instrument itself upon demand or in an external conversion program.

The most common types of data produced by power quality monitoring devices include the following:

- Voltage and current waveform captures (e.g., transients, waveform distortion)
- Voltage and current phasor captures (e.g., short and long duration RMS variations)
- Steady state value logs (e.g., true power factor at 15-minute intervals for some period of time)

PQDIF represents these data types by establishing a few basic concepts and categorization levels. The categorization levels are as follows:

- Quantity type
- Quantity measured

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

— Quantity characteristic

— Phase identification

These categories uniquely identify the primary data series of a logical channel that represents a specific measured or simulated quantity. As this document ~~we~~ will show later, a logical channel can contain multiple data series that can be further qualified with attributes such as the following:

— Minimum value over an interval

— Maximum value over an interval

— Average value over an interval

— Instantaneous value

— Phase angle of a complex value~~.~~

A waveform, or RMS trace, or value can be recorded due to ~~as a result of~~ a triggered event as defined by IEEE Std 1159~~-1995~~[3], a periodic capture programmed into an instrument, a user-initiated manual trigger, or some external trigger based on a contact closure or algorithm evaluation. PQDIF provides the means to identify these possible states as well as the IEEE ~~Std~~ 1159~~-1995~~ disturbance categorization of triggered events if appropriate.

The data associated with these data types are represented in an observation record. The format of that data is defined in the most recent data source record. Further information about the state of the instrument or simulation program that created the observation record is optionally contained within the most recent monitor settings record.

For triggered events, all ~~of~~ the data associated with a disturbance as defined in IEEE Std 1159~~-1995~~ is contained within a single observation. For example, an RMS variation is defined to begin when one or more phases of voltage goes outside of threshold and is defined to end when all phases are back within the threshold. All~~-of~~ the channels used to represent the RMS trace(s) and/or waveform(s) captured during that disturbance would be stored in a single PQDIF observation record.

[3] ~~Information on references can be found in Clause 2.~~

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

7.1 General procedures

Translation of power quality data of any format requires that the following information be gleaned from the native file format and translated to PQDIF:

a) Measured data type—Description of what is represented by the actual quantitative measurement data. This data source information associated with a given numeric measurement data set is often referred to as the data channel definition.

b) Monitor information—General monitor information such as threshold settings, installation date, calibration information, etc ~~and so on~~. The extent of monitor information that needs to be translated to PQDIF is dependent on the destination of the translated PQDIF files—~~,~~ the final application that will use the PQDIF files as an input.

c) Measured data values—Actual quantitative measurement data associated with a defined data channel.

The following subclauses ~~subsections~~ discuss the PQDIF element tags that must be populated to relate the three pieces of information listed. The next subclause ~~"Channel Specifications Overview" section~~ discusses the required tags for specifying data channels (measurement data type) in the data source record. Subclause 7.1.2 ~~The "Monitor Settings Specifications Overview"~~ discusses the required tags for specifying monitor information in the optional monitor settings record. Finally, 7.1.3 ~~the "Observation Specifications Overview"~~ discusses the required tags for specifying measurement values recorded for a specified channel (measurement data values) in the observation record.

7.1.1 Channel specifications overview

Before the individual measurements can be translated to the PQDIF format, individual logical channels must be defined in the PQDIF file that relate what the measured data from the instrument's physical channels represents. A logical channel in PQDIF is a unique combination of quantity type, quantity measured, quantity characteristic, and phase. The data source record is used in PQDIF to define the attributes of all ~~of~~ the logical channels that may be referenced by the instance data in a PQDIF file.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

PQDIF requires several data fields to completely describe these elements that define each channel. Table 4 lists the required element tags that must be assigned values for each channel. Table 4 shows two series definitions. Note that a channel may consist of only one series, but most channels consist of two (e.g., a series of time values and a parallel series of magnitude values representing the instantaneous value of a waveform at each time point in the time series). Also, it should be noted that the tags listed in Table 4 are only the required tags. There are many other optional tags that may be assigned values.

Each of the required element tags listed in Table 4 have ~~has~~ an associated set of acceptable values. The "Example Value" column of Table 4 shows the values that should be assigned to define the channel that records instantaneous voltage waveforms on phase A-neutral. See Annex B for the list of acceptable values for each of the element tags defined in Table 4.

Table 4 —Required data source record element tags

Level	Element tag	Example value
0	tagRecDataSource	--NA--
1	tagNominalFrequency	60.0
1	tagChannelDfns	--NA--
2	tagOneChannelDfn	--NA--
3	tagPhaseID	*ID_Phase_AN*
3	tagQuantityTypeID	*ID_QT_Waveform*
3	tagQuantityMeasuredID	*ID_QM_Voltage*
3	tagSeriesDfns	--NA--
4	tagOneSeriesDfn	--NA--
5	tagValueTypeID	*ID_Series_Value_Type_Time*
5	tagQuantityUnitsID	*ID_QU_Seconds*
5	tagQuantityCharacteristicID	*ID_QC_None*
5	tagStorageMethodID	*ID_Series_Method_Values*
4	tagOneSeriesDfn	--NA--
5	tagValueTypeID	*ID_Series_Value_Type_Val*
5	tagQuantityUnitsID	*ID_QU_Volts*
5	tagQuantityCharacteristicID	*ID_QC_Instantaneous*
5	tagStorageMethodID	*ID_Series_Method_Values*

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

7.1.2 Monitor settings specifications overview

The tags in the monitor settings record provide basic, vendor-independent instrument settings information. The monitor settings record is not required. If, however, information such as the trigger values for a given channel is critical to the data being translated, it is recommended that the monitor settings record be created. Most of the defined tags are optional and are to be populated only if the data exists ~~exist~~ and would be useful to other applications.

Table 5 lists the required monitor settings tags and example values for the example channel defined in Table 4. Note that only one channel is included in Table 5, but monitor setting tag values should be specified for each channel defined. Furthermore, there are many channel level (level 3) tags that are optional if the information is available.

Table 5 —Required monitor settings record element tags

Level	Element tag	Example value
0	TagRecMonitorSettings	--NA--
1	TagEffective	Date settings were implemented.
1	TagTimeInstalled	Date monitored installed
1	TagChannelSettingsArray	--NA--
2	TagOneChannelSetting	--NA--
3	TagChannelDefnIdx	57
3	TagTriggerTypeID	--NA--

7.1.3 Observation specifications overview

Assignment of the observation element tags is primarily a matter of listing the measurement values in a way that coordinates with the way ~~manner in which~~ the associated channel has been specified. Note that a single PQDIF observation may consist of many different measurements on one or more physical channels that are associated with the same trigger. For example, a single voltage sag might trigger both waveform and rms phasor measurements. Although the data recorded due to this single sag is ~~are~~ associated with many different channels, it is more efficient to store all ~~of~~ the data associated with a single trigger in a single observation. This also allows for proper association of data channels in the final application that~~, which~~ uses the PQDIF files.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

Table 6 lists the required element tags that define a given observation. As with the channel specification, there are other optional tags that can be assigned as well. Permissible values for each element tag listed in Table 6 are provided in PQDIF logical structure document. The "Example Value" column of Table 6 lists possible values for a Phase A voltage waveform associated with the channel specified in Table 4.

Table 6 —Required observation record element tags

Level	Element tag	Example value
0	TagRecObservation	--NA--
1	TagObservationName	*Phase A Voltage Waveform*
1	TagTimeCreate	*12/9/1998 11:42:24.453813818*
1	TagTimeStart	*2/13/1998 23:17:44.086079597*
1	TagTriggerMethodID	*ID_Trigger_Meth_Channel*
1	TagTimeTriggered	*2/13/1998 23:17:44.086079597*
1	TagChannelInstances	--NA--
2	TagOneChannelInst	--NA--
3	TagChannelDefnIdx	57
3	TagSeriesInstances	--NA--
4	TagOneSeriesInstance	--NA--
5	TagSeriesValues	0,0.00013,0.00026, . . .
4	TagOneSeriesInstance	--NA--
5	TagSeriesValues	-4607.4621, -5030.9689, -5428.9193, . . .

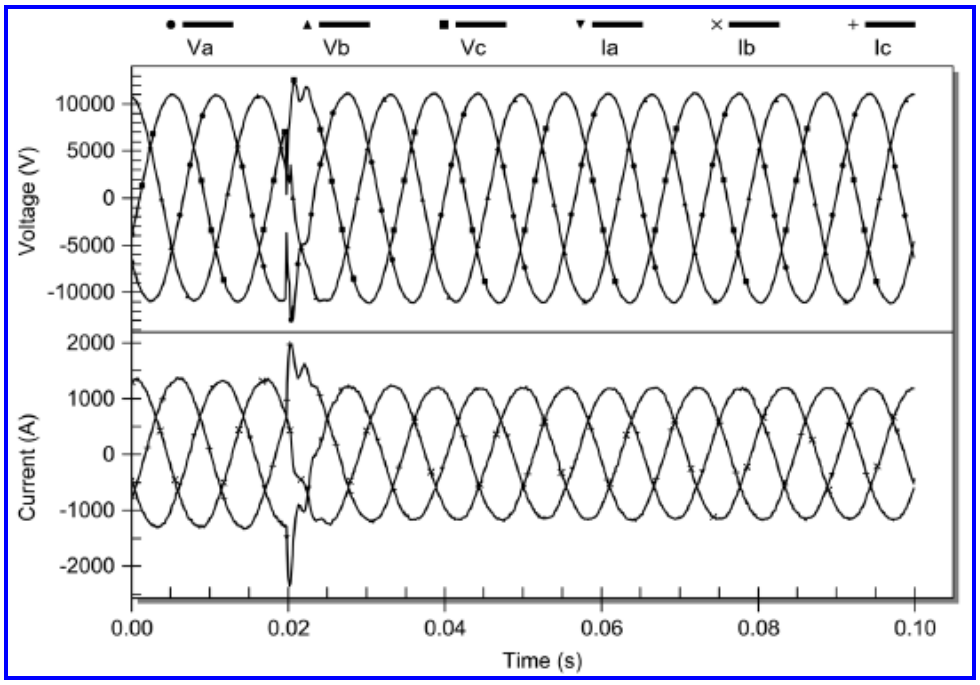
The following subclauses ~~Clause 7.2 through 7.4~~ describe how to represent these common power quality data types in PQDIF by example.

7.2 Representing a waveform recording

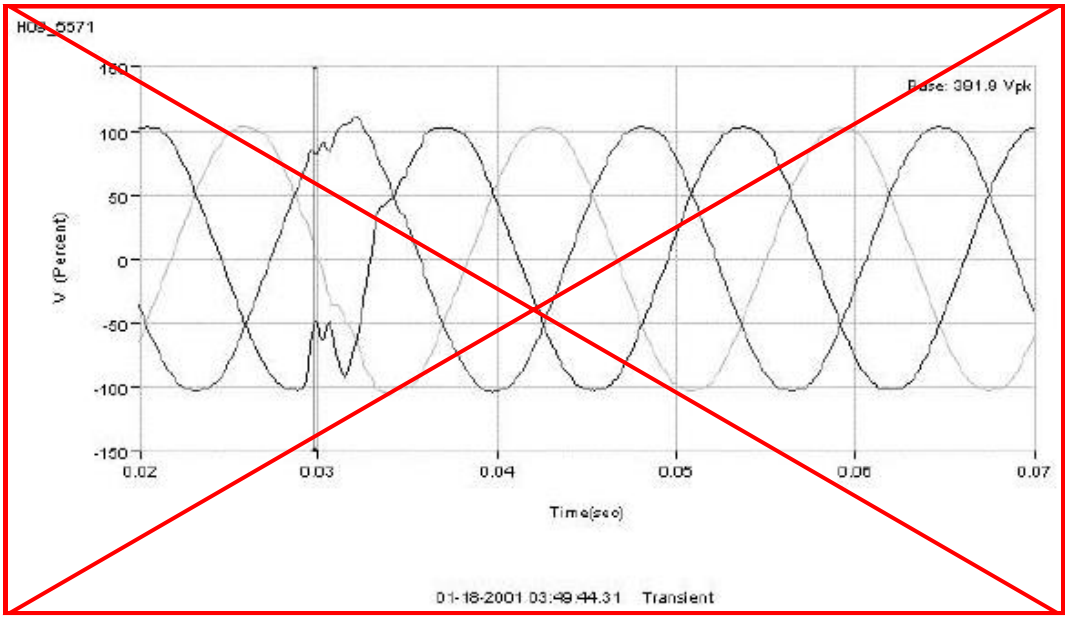
A waveform recording is represented in PQDIF using channel definitions of the QT_WAVEFORM quantity type. This is the simplest of the PQDIF quantity types and one of the most commonly used types for instruments that capture waveforms of voltage and current. Figure 8 illustrates a transient event captured by a power monitor, exported into the PQDIF representation described here and visualized by a PQDIF viewer program.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

Figure 8—Low-frequency transient (capacitor switching)



(New)



~~(Deleted)~~

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

The individual measurements logged by the instrument are stored in PQDIF "observation" records. If a single trigger causes a power quality monitor to save ~~results in saving~~ waveforms for many ~~a number of~~ different quantities (for example, cross-triggered measurements), it is useful for all ~~of~~ the measurements associated with the same trigger to be included in a single observation record. If an instrument can capture more than one waveform block during a single disturbance~~,~~ as defined by IEEE Std1159 ~~1995~~ (e.g., 5 ~~five~~ cycles of waveform at the beginning of a sag and 5 ~~five~~ cycles of waveform at the end of the same sag), then more than one instance of a specific ~~particular~~ channel definition may be recorded in the PQDIF observation.

The following text is an ASCII dump from a PQDIF utility program that shows the top-level tags of a data source record for a real PQDIF file. These tags permit the specification of basic information about the data source.

```
-Collection -- tag: tagRecDataSource (level 0)
 | The checksum for this record is correct.
 +-Scalar -- tag: tagDataSourceTypeID (type: GUID)
 | value: {e6b51730-f747-11cf-9d890080} - ID_DS_TYPE_MEASURE
 +-Scalar -- tag: tagVendorID (type: GUID)
 | value: {e6b5170a-f747-11cf-9d890080} - ID_VENDOR_ELECTROTEK
 +-Scalar -- tag: tagEquipmentID (type: GUID)
 | value: {e6b51721-f747-11cf-9d890080} - ID_EQUIP_ETK_TESTPROGRAM
 +-Vector -- tag: tagSerialNumberDS (type: CHAR1) [ 4 ]
```

(New)

~~-Collection -- tag: tagRecDataSource (level 0)~~
~~| The checksum for this record is correct.~~
~~+-Scalar -- tag: tagDataSourceTypeID (type: GUID)~~
~~| value: {e6b51730-f747-11cf-9d890080} - ID_DS_TYPE_MEASURE~~
~~+-Scalar -- tag: tagVendorID (type: GUID)~~
~~| value: {e6b5170a-f747-11cf-9d890080} - ID_VENDOR_ELECTROTEK~~
~~+-Scalar -- tag: tagEquipmentID (type: GUID)~~
~~| value: {e6b51721-f747-11cf-9d890080} - ID_EQUIP_ETK_TESTPROGRAM~~
~~+-Vector -- tag: tagSerialNumberDS (type: CHAR1) [4]~~
~~| value: '1.0'~~
~~+-Vector -- tag: tagVersionDS (type: CHAR1) [4]~~
~~| value: '1.0'~~
~~+-Vector -- tag: tagNameDS (type: CHAR1) [14]~~
~~| value: 'PQDIF Convert'~~
~~+-Vector -- tag: tagOwnerDS (type: CHAR1) [4]~~
~~| value: 'EWG'~~
~~+-Vector -- tag: tagLocationDS (type: CHAR1) [1]~~

~~(Deleted)~~

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

```
| value: '1.0'
+-Vector -- tag: tagVersionDS (type: CHAR1) [ 4 ]
| value: '1.0'
+-Vector -- tag: tagNameDS (type: CHAR1) [ 14 ]
| value: 'PQDIF Convert'
+-Vector -- tag: tagOwnerDS (type: CHAR1) [ 4 ]
| value: 'John Smith'
+-Vector -- tag: tagLocationDS (type: CHAR1) [ 1 ]
| value: ''
+-Vector -- tag: tagTimeZoneDS (type: CHAR1) [ 1 ]
| value: ''
```

(New)

~~| value: ''~~
~~+-Vector -- tag: tagTimeZoneDS (type: CHAR1) [1]~~
~~| value: ''~~

~~(Deleted)~~

The following ASCII dump from the same PQDIF file shows the channel definition for one phase of a voltage channel waveform channel:

```
+-Collection -- tag: tagChannelDefns (level 1)
 | +-Collection -- tag: tagOneChannelDefn (level 2)
 | | +-Vector -- tag: tagChannelName (type: CHAR1) [ 13 ]
 | | | value: 'Waveform VAB'
 | | +-Scalar -- tag: tagPhaseID (type: UNS_INTEGER4)
 | | | value: 1 - ID_PHASE_AB
 | | +-Scalar -- tag: tagQuantityMeasuredID (type: UNS_INTEGER4)
 | | | value: 1 - ID_QM_VOLTAGE
 | | +-Scalar -- tag: tagQuantityTypeID (type: GUID)
 | | | value: {67f6af80-f753-11cf-9d890080} - ID_QT_WAVEFORM
 | | +-Collection -- tag: tagSeriesDefns (level 3)
 | | | +-Collection -- tag: tagOneSeriesDefn (level 4)
 | | | | +-Scalar -- tag: tagQuantityUnitsID (type: UNS_INTEGER4)
 | | | | | value: 2 - ID_QU_SECONDS
 | | | | +-Scalar -- tag: tagQuantityCharacteristicID (type: GUID)
 | | | | | value: {a6b31ae5-b451-11d1-ae170060} - ID_QC_NONE
 | | | | +-Scalar -- tag: tagValueTypeID (type: GUID)
 | | | | | value: {c690e862-f755-11cf-9d890080} - ID_SERIES_VALUE_TYPE_TIME
 | | | | +-Scalar -- tag: tagStorageMethodID (type: UNS_INTEGER4)
 | | | | | value: 4 - ID_SERIES_METHOD_INCREMENT
 | | | | +-(End of collection)
 | | | +-Collection -- tag: tagOneSeriesDefn (level 4)
 | | | | +-Scalar -- tag: tagQuantityUnitsID (type: UNS_INTEGER4)
 | | | | | value: 6 - ID_QU_VOLTS
 | | | | +-Scalar -- tag: tagQuantityCharacteristicID (type: GUID)
 | | | | | value: {a6b31add-b451-11d1-ae170060} - ID_QC_INSTANTANEOUS
 | | | | +-Scalar -- tag: tagValueTypeID (type: GUID)
 | | | | | value: {67f6af97-f753-11cf-9d890080} - ID_SERIES_VALUE_TYPE_VAL
 | | | | +-Scalar -- tag: tagStorageMethodID (type: UNS_INTEGER4)
 | | | | | value: 3 - ID_SERIES_METHOD_VALUES | ID_SERIES_METHOD_SCALED
 | | | | +-Scalar -- tag: tagSeriesNominalQuantity (type: REAL8)
 | | | | | value: 48083.261121
 | | | | +-(End of collection)
 | | | +-(End of collection)
 | | +-(End of collection)
```

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

This channel definition specifies a logical channel of type QT_WAVEFORM and is associated with phase A to phase B voltage. ~~By definition,~~ A QT_WAVEFORM channel contains two series definitions—~~:~~ a time series and a value series. The time series is of type ID_SERIES_VALUE_TYPE_TIME using the ID_SERIES_METHOD_INCREMENT storage method. This means that the time series is stored as a relative time (relative to the time stamp of an instantiated channel in an observation) using a simple starting time, increment~~,~~ and count method. This is efficient for instruments that take point-on-wave samples with a fixed sampling rate.

The value series is of type ID_SERIES_VALUE_TYPE_VAL using the ID_SERIES_METHOD_SCALED | ID_SERIES_METHOD_VALUES storage method, and ~~it~~ represents the QC_INSTANTANEOUS (point-on-wave) characteristic of the voltage. The storage method is the result of a logical OR~~'d value~~ of scaled and values series tags to indicate that the waveform voltage values in an instance of this channel will be represented as an array of integers to which a scale and offset are applied to obtain the original engineering units of the data.

A nominal value is set for this channel for the benefit of reader programs. This is most often used in a reader program to permit displaying the resulting waveform in percent or per-unit.

This channel definition was the fourth channel definition in the data source record. The zero-based index of 3 is used in the instance data in an observation to refer to this definition.

The following text shows the ASCII dump of the tags in the monitor settings record in this same PQDIF file:

```
-Collection -- tag: tagRecMonitorSettings (level 0)
 | The checksum for this record is correct.
 +-Scalar -- tag: tagEffective (type: TIMESTAMPPQDIF)
 | value: 7/6/1999 11:40:45.999999926
 +-Scalar -- tag: tagTimeInstalled (type: TIMESTAMPPQDIF)
 | value: 1/1/1982 0:0:0.000000000
 +-Scalar -- tag: tagTimeRemoved (type: TIMESTAMPPQDIF)
 | value: 1/1/2020 0:0:0.000000000
 +-Scalar -- tag: tagUseCalibration (type: BOOLEAN4)
 | value: 'FALSE'
 +-Scalar -- tag: tagUseTransducer (type: BOOLEAN4)
 | value: 'FALSE'
```

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

This monitor settings record simply specifies the time that the instrument was installed, the effective time of the monitor settings record, and whether any calibration or transducer settings in the monitor settings record are to be used in combination with the instance data in observation records to obtain the original measured values. Most often, transducer and calibration info ~~information~~ is stored in the monitor settings record for reference purposes~~,~~ but the instance data already has ~~have~~ these values considered ~~taken into account~~ in the data values reported there. This file therefore represents the default of FALSE for both entries, and in fact, this file does not even specify the optional tags for transducer and calibration info ~~information~~.

The following text shows the ASCII dump of the top-level tags in an observation record containing a transient waveform recording in this same PQDIF file:

```
-Collection -- tag: tagRecObservation (level 0)
 | The checksum for this record is correct.
 +-Vector -- tag: tagObservationName (type: CHAR1) [ 4 ]
 | value: 'Obs'
 +-Scalar -- tag: tagTimeCreate (type: TIMESTAMPPQDIF)
 | value: 7/7/1999 20:40:58.000000217
 +-Scalar -- tag: tagTimeStart (type: TIMESTAMPPQDIF)
 | value: 7/6/1999 11:40:45.999999926
 +-Scalar -- tag: tagTriggerMethodID (type: UNS_INTEGER4)
 | value: 1 - ID_TRIGGER_METH_CHANNEL
 +-Scalar -- tag: tagTimeTriggered (type: TIMESTAMPPQDIF)
 | value: 7/6/1999 11:40:46.030793991
 +-Vector -- tag: tagChannelTriggerIdx (type: UNS_INTEGER4) [ 1 ]
 | values: 4
 +-Collection -- tag: tagChannelInstances (level 1)
 | +-Collection -- tag: tagOneChannelInst (level 2)
 | | +-Scalar -- tag: tagChannelDefnIdx (type: UNS_INTEGER4)
 | | | value: 3
 | | +-Collection -- tag: tagSeriesInstances (level 3)
 | | | +-Collection -- tag: tagOneSeriesInstance (level 4)
 | | | | +-Vector -- tag: tagSeriesValues (type: REAL4) [ 3 ]
 | | | | +-(End of collection)


| | | +-Collection -- tag: tagOneSeriesInstance (level 4)
| | | | +-Vector -- tag: tagSeriesValues (type: INTEGER2) [ 1536 ]
| | | | +-Scalar -- tag: tagSeriesBaseQuantity (type: REAL8)
| | | | | value: 34000.000000
| | | | +-Scalar -- tag: tagSeriesScale (type: REAL8)
| | | | | value: 1.568127
| | | | +-Scalar -- tag: tagSeriesOffset (type: REAL8)
| | | | | value: 0.000000
| | | | +-(End of collection)
| | | +-(End of collection)
| | +-(End of collection)
```

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

The tags in this observation specifies ~~specify~~ an optional name "Obs," specifies the time of creation of the record, the absolute time of the first data point in the data, and the time (if applicable) of the trigger that resulted in this observation being saved. Since ~~As~~ this is a triggered event, the trigger method tag is specified to be of type channel indicating that a specific channel-based algorithm triggered the observation to be recorded. Additionally, the channel index array specifies which channel caused the trigger (channel index 4—~~,~~—phase B to C voltage—~~,~~—in this case). The tagChannelDefnIdx indicates that this ~~these~~ instance data is ~~are~~ for the fourth channel definition (zero based index of 3) within the data source record~~,~~ as shown earlier.

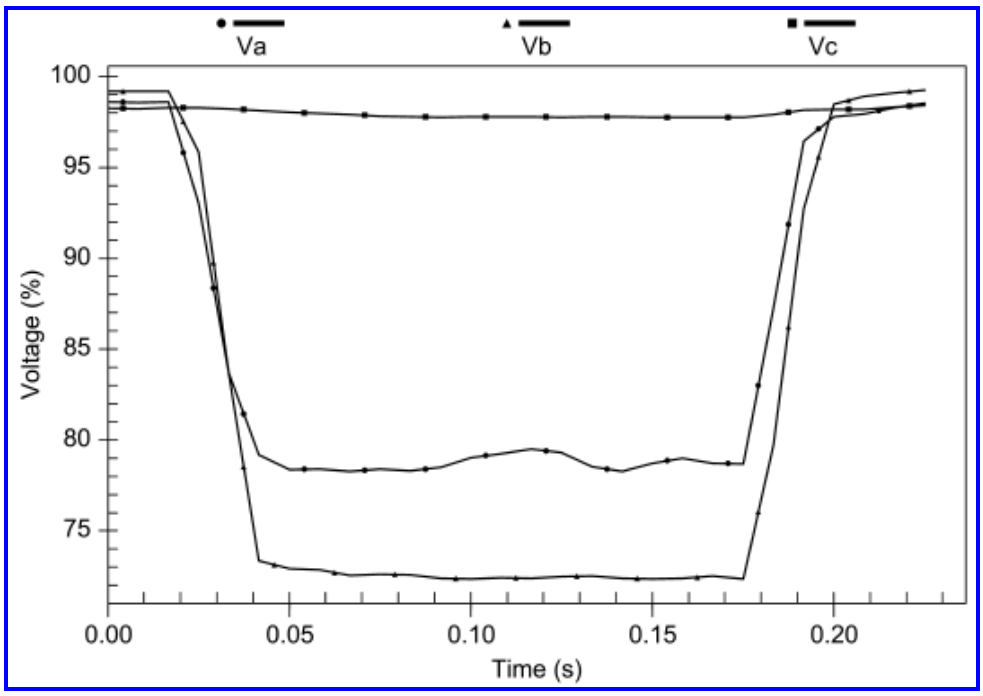
The instance data itself contains ~~contain~~ the data for both the time and the value series. The time series is first and consists of an array of three 4-byte floating point numbers representing the number of increment/count pairs (in this case~~,~~ 1), count~~,~~ and increment of the time series. The value series consists of an array of 1536 two-~~,~~ ~~2~~ byte integers. This array is scaled by the tagSeriesScale factor and then offset by the tagSeriesOffset factor to get the engineering units for the recorded waveform points.

7.3 Representing a ~~an~~ RMS variation recording

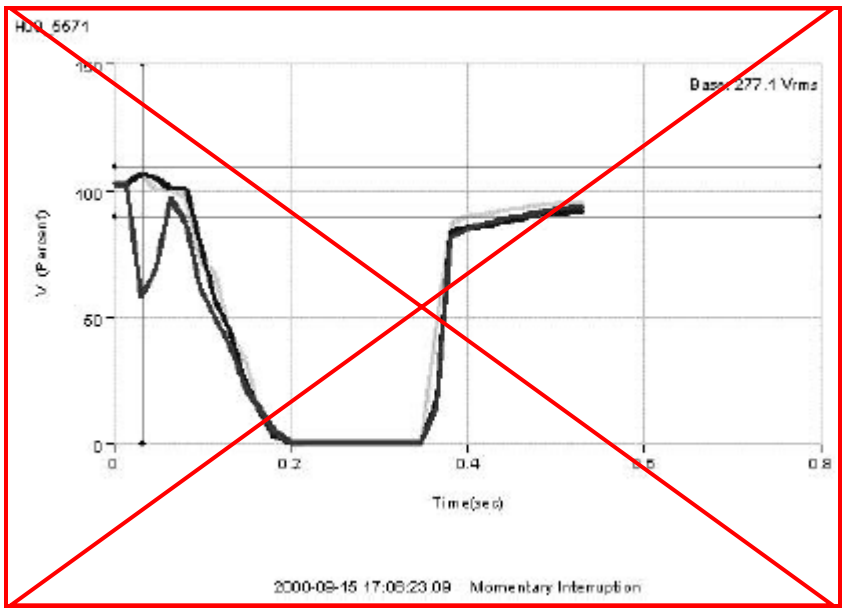
The RMS trace for an RMS variation disturbance is represented in PQDIF using channel definitions of the QT_PHASOR quantity type. There is another type that will be discussed later (QT_VALUELOG) that could be used for this purpose, but the QT_PHASOR type has been reserved for this purpose by convention. This type can be used to store the ~~is capable of~~ recording the instantaneous phasor representation of the RMS value of voltage or current.

Most instruments that gather this type of information use a one-cycle window to calculate the RMS value of the signal~~,~~ and some can ~~are able to~~ calculate the phase angle relative to a reference phase. These instruments can make this measurement as fast as their fundamental A/D resolution allows (e.g., every 1/256 of a cycle) or~~,~~ more commonly, once or twice a cycle. The QT_PHASOR type is ideally suited for representing this information. Figure 9 illustrates an RMS variation event captured by a three-phase power monitor, exported to PQDIF, and then visualized in a PQDIF viewer program.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019



(New)



~~(Deleted)~~

Figure 9—RMS variation showing all three phases~~ and trigger lines~~

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

The QT_PHASOR type and the QT_VALUELOG type to be discussed later have more options for series definitions than the QT_WAVEFORM type. QT_PHASOR uses a time series as its first series as before, but more than one value series is possible. These variants allow series to be defined to hold minimum, maximum, average, and present values over an interval for a variable. This capability is used by those instruments and simulation programs that change their recording rate for a long disturbance. These data sources typically still track the RMS value on a cycle-by-cycle of faster data rate internally and use that information to maintain max ~~maximum~~, min ~~minimum~~, and average registers for a value. Instead of saving this instantaneous data, however, they save the min ~~minimum~~, max ~~maximum~~, and average values at some prescribed rate (e.g., every 6 ~~six~~ cycles), and then reset the min ~~minimum~~, max ~~maximum~~, and average registers for the next interval.

The following text is an ASCII dump from a PQDIF utility program that shows the channel definition for one phase of a voltage channel representing recordings of type QT_PHASOR for phase A to B voltage:

```
+-Collection -- tag: tagOneChannelDefn (level 2)
| | +-Vector -- tag: tagChannelName (type: CHAR1) [ 11 ]
| | | value: 'Phasor VAB'
| | +-Scalar -- tag: tagPhaseID (type: UNS_INTEGER4)
| | | value: 5 - ID_PHASE_AB
| | +-Scalar -- tag: tagQuantityMeasuredID (type: UNS_INTEGER4)
| | | value: 1 - ID_QM_VOLTAGE
| | +-Scalar -- tag: tagQuantityTypeID (type: GUID)
| | | value: {67f6af81-f753-11cf-9d890080} - ID_QT_PHASOR
| | +-Collection -- tag: tagSeriesDefns (level 3)
| | | +-Collection -- tag: tagOneSeriesDefn (level 4)
| | | | +-Scalar -- tag: tagQuantityUnitsID (type: UNS_INTEGER4)
| | | | | value: 2 - ID_QU_SECONDS
| | | | +-Scalar -- tag: tagQuantityCharacteristicID (type: GUID)
| | | | | value: {a6b31ae5-b451-11d1-ae170060} - ID_QC_RMS
| | | | +-Scalar -- tag: tagValueTypeID (type: GUID)
| | | | | value: {c690e862-f755-11cf-9d890080} - ID_SERIES_VALUE_TYPE_TIME
| | | | +-Scalar -- tag: tagStorageMethodID (type: UNS_INTEGER4)
| | | | | value: 1 - ID_SERIES_METHOD_VALUES
| | | | +-(End of collection)
| | | +-Collection -- tag: tagOneSeriesDefn (level 4)
```

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019

```
| | | | +-Scalar -- tag: tagQuantityUnitsID (type: UNS_INTEGER4)
| | | | | value: 6 - ID_QU_VOLTS
| | | | +-Scalar -- tag: tagQuantityCharacteristicID (type: GUID)
| | | | | value: {a6b31ae5-b451-11d1-ae170060} - ID_QC_RMS
| | | | +-Scalar -- tag: tagValueTypeID (type: GUID)
| | | | | value: {67f6af98-f753-11cf-9d890080} - ID_SERIES_VALUE_TYPE_MIN
| | | | +-Scalar -- tag: tagStorageMethodID (type: UNS_INTEGER4)
| | | | | value: 3 - ID_SERIES_METHOD_VALUES | ID_SERIES_METHOD_SCALED
| | | | +-Scalar -- tag: tagSeriesNominalQuantity (type: REAL8)
| | | | | value: 34000.000000
| | | | +-(End of collection)
| | | +-Collection -- tag: tagOneSeriesDefn (level 4)
| | | | +-Scalar -- tag: tagQuantityUnitsID (type: UNS_INTEGER4)
| | | | | value: 6 - ID_QU_VOLTS
| | | | +-Scalar -- tag: tagQuantityCharacteristicID (type: GUID)
| | | | | value: {a6b31ae5-b451-11d1-ae170060} - ID_QC_RMS
| | | | +-Scalar -- tag: tagValueTypeID (type: GUID)
| | | | | value: {67f6af99-f753-11cf-9d890080} - ID_SERIES_VALUE_TYPE_MAX
| | | | +-Scalar -- tag: tagStorageMethodID (type: UNS_INTEGER4)
| | | | | value: 3 - ID_SERIES_METHOD_VALUES | ID_SERIES_METHOD_SCALED
| | | | +-Scalar -- tag: tagSeriesNominalQuantity (type: REAL8)
| | | | | value: 34000.000000
| | | | +-(End of collection)
| | | +-Collection -- tag: tagOneSeriesDefn (level 4)
| | | | +-Scalar -- tag: tagQuantityUnitsID (type: UNS_INTEGER4)
| | | | | value: 6 - ID_QU_VOLTS
| | | | +-Scalar -- tag: tagQuantityCharacteristicID (type: GUID)
| | | | | value: {a6b31ae5-b451-11d1-ae170060} - ID_QC_RMS
| | | | +-Scalar -- tag: tagValueTypeID (type: GUID)
| | | | | value: {67f6af9a-f753-11cf-9d890080} - ID_SERIES_VALUE_TYPE_AVG
| | | | +-Scalar -- tag: tagStorageMethodID (type: UNS_INTEGER4)
| | | | | value: 3 - ID_SERIES_METHOD_VALUES | ID_SERIES_METHOD_SCALED
| | | | +-Scalar -- tag: tagSeriesNominalQuantity (type: REAL8)
| | | | | value: 34000.000000
| | | | +-(End of collection)
| | | +-Collection -- tag: tagOneSeriesDefn (level 4)
| | | | +-Scalar -- tag: tagQuantityUnitsID (type: UNS_INTEGER4)
| | | | | value: 17 - ID_QU_DEGREES
| | | | +-Scalar -- tag: tagQuantityCharacteristicID (type: GUID)
| | | | | value: {a6b31add-b451-11d1-ae170060} - ID_QC_INSTANTANEOUS
| | | | +-Scalar -- tag: tagValueTypeID (type: GUID)
| | | | | value: {3d786f9d-f76e-11cf-9d890080} -
ID_SERIES_VALUE_TYPE_PHASEANGLE
| | | | +-Scalar -- tag: tagStorageMethodID (type: UNS_INTEGER4)
| | | | | value: 3 - ID_SERIES_METHOD_VALUES | ID_SERIES_METHOD_SCALED
| | | | +-(End of collection)
| | | +-(End of collection)
| | +-(End of collection)
```

This channel definition specifies a logical channel of type QT_PHASOR and is associated with phase A to phase B voltage. As mentioned earlier, the QT_PHASOR channel can contain multiple series definitions. In this case, series definitions are provided for minimum, maximum, average, and phase angle values from the recording. The time series is of type ID_SERIES_VALUE_TYPE_TIME using the ID_SERIES_METHOD_VALUES storage method. The implementers of this PQDIF file chose to use values instead of increments since ~~incremental because~~ this instrument can report its RMS recordings with an irregular time base.

Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2026 at 11:03:40 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 1159.3-2019