REVISION OF IEEE STD 1159.3 PQDIF

Daniel SABIN
Electrotek Concepts – USA
d.sabin@ieee.org

William W. DABBS
Electrotek Concepts – USA
bdabbs@electrotek.com

ABSTRACT

IEEE Std 1159.3-2003 specifies the Power Quality Data Interchange Format, which is better known as PQDIF. This binary file format is used for the transfer of power quality data between software and hardware systems. 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. Although it was reaffirmed in 2008, the field of power quality monitoring has evolved in the past decade, and the changes in this domain are not reflected in the current version of 1159.3. The paper will present the changes planned for the revision that will take place in 2015 and 2016.

INTRODUCTION

Over the last thirty years, many power quality monitoring instruments have been employed in the collection of power quality measurements from tens, hundreds, and sometimes thousands of monitoring points in transmission, distribution, and end-user systems. The reasons for monitoring vary, and consequently, so does the structure of data contained in these measurements. IEEE Std 1159.3-2003 PQDIF file format [2] provides a compact, flexible, extensible means to exchange these diverse measurements between software applications. PQDIF is an acronym for Power Quality Data Interchange Format.

The Transmission & Distribution Committee of the IEEE Power & Energy Society is sponsoring a revision of IEEE 1159.3. The purpose of this proposed paper is to share an overview of the existing standard with the CIRED community, along with the end-user tools available for viewing PQDIF files, such as the IEEE PQDIF File Utility, PQDIFFR2, and PQDiffractor. The paper will also summarize the free software libraries maintained by the task force for reading and writing PQDIF files. These libraries are useful in native C++, .NET, and ActiveX/COM applications.

The paper will present the changes planned for the revision that will take place from 2015 and 2016. For example, it adds recommendations on how to store rapid voltage change (RVC) events or their characteristics to the existing IEEE 1159.3 standard.

OVERVIEW OF IEEE 1159.3 PQDIF

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.

Physical Structure

PQDIF employs a relatively simple, hierarchical physical structure. At the highest level, a file consists of one or more records:

```
Record 1
Record 2
Record 3
…
Record N
```

Each record is composed of a collection of elements. Each element is one of three types: a scalar (single value), a vector (series of values, or array), or a collection of elements.

```
Record
    Collection 1
            Scalar 1
            Scalar 2
            Vector 1
            Vector 2
            Collection 2
                    Scalar 3
                    Scalar 4
            Collection 3
                    Scalar 5
                    Scalar 6
```

Scalar and vector element values are two of a fixed set of types. The complete list of types is presented in Table 1.

Table 1: Intrinsic PQDIF Data Types

Name	Description
BOOL1	1 byte Boolean
BOOL2	2 byte Boolean
BOOL4	4 byte Boolean
CHAR1	ASCII character
CHAR2	Unicode character
INT1	1 byte integer
INT2	2 byte integer
INT4	4 byte integer
UINT1	1 byte unsigned integer
UINT2	2 byte unsigned integer
UINT4	4 byte unsigned integer
REAL4	4 byte real
REAL8	8 byte real
COMPLEX8	Complex 4 byte real pair
COMPLEX16	Complex 8 byte real pair
DT	Time stamp with microsecond resolution
GUID	16 byte globally unique identifier

Complex data elements are created using collections of elements. Since a collection can contain nested collections, any level of complexity can be obtained without the need to add new intrinsic data types. This is the mechanism that PQDIF uses to achieve flexibility while avoiding the need for constant revision.

This simple set of rules defines the physical structure of all PQDIF files. Once an application implements them it can parse any present or future PQDIF file.

Logical Structure

The logical structure defines the rules for how records and record elements are organized to represent power quality data. The low-level details of these records are beyond the scope of this paper; however, a high level understanding of the logical structure is necessary in order to appreciate the organization of the XML representation of a PQDIF file. The essential concepts that must be understood are tags and IDs, channels, and record types.

Tags and Ids

Tags and IDs are the foundation of the logical structure of PQDIF. Every PQDIF element has a tag that identifies the meaning of the element. For example, the primary collection of each record has a tag identifying the record type. Ids are the range of values that a tag may have. The PQDIF specifications define a standard set of tags and for each tag, its associated element type, its place in the element hierarchy of a record, and its range of ID values.

Each tag value is implemented using a globally unique identifier (GUID). Id values can be optionally implemented as GUID or integer values as appropriate. Since a unique GUID can be generated without consulting a central authority, a PQDIF application can create private tags, only understood by that application. Since all PQDIF applications are required to ignore elements with tags that they do not understand, these private tags have no effect on other applications that parse the PQDIF file. This mechanism ensures that new tags can be added to the PQDIF standard without breaking existing applications. GUID based tag and ID values are the part of the extensible nature of PQDIF.

Channels

A channel represents one of a set of data streams that are generated by a monitor. Each channel should have unique quantity (e.g., rms voltage) and phase (e.g., phase A-N) information. Channels are created for both real measurements (e.g., rms voltage and rms current) and derived measurements (e.g., voltage total harmonic distortion, Pst).

Data Source Records

Data source records contain collections of definitions for the channels of a monitor. These channel definition collections contain elements that identify the quantity, phase, units, etc. of a channel. The values contained in a channel definition do not change over time. Many observation records share a single data source record. This avoids replicating definition information in each observation record.

Monitor Setting Records

Like data source records, monitor settings records contain collections that are referenced by the channel instance collections of many observation records. Rather than defining the unchanging values of a channel definition, channel settings collections contain elements to store information about the parameters that are in effect when a measurement is captured. For example, trigger threshold might be recorded in a settings record. Monitor settings records avoid the need to replicate channel definitions elements when a settings change is made either by an operator or automatically by the monitor.

Observation Records

Observation records contain the actual measured values. Each observation record contains one or more channel instance collection. These collections reference channel definition collections and channel settings collections in the associated data source and monitor settings records, respectively. Each channel collection typically contains a vector of time values and one or more vectors of the channel quantity values. For example, separate vectors could be used for average, minimum, and maximum values.

Data source, monitor settings, and observation records are associated by their order in a PQDIF file. The data source record must precede the monitor settings and observation records that reference it. Similarly, a monitor settings record must precede the observation records that reference it.

SOFTWARE RESOURCES

The IEEE P1159.3 Working Group maintains a number of software applications and source code that can be used to read and write PQDIF files. They can be downloaded from the working group's website:
http://grouper.ieee.org/groups/1159/3/

Source Code Header Files

These files are the core of the specification. They contain all of the tags listed in the normative annexes in the standard. These files are updated periodically as vendors add new tags. The updates are always backward compatible.

IEEE PQDIF Libraries

The working group's web site also includes the source code in C++ Libraries for reading and writing PQDDIF files. These files have been used as the source code for many Microsoft Windows and UNIX implementations of PQDIF.

PQDIF COM Library

PQDCOM4.DLL is an ActiveX/COM Library for reading and writing PQDIF files used by many third-parties on legacy Microsoft Windows applications. It is intended for use with Microsoft Visual Basic 6, Microsoft Office VBA, MATLAB, etc. This library is free to use and distribute and its source code is available on the working group's web site.

PQDIF .NET Library

PQDIFNET.DLL is a .NET Assembly for reading and writing PQDIF Files in modern Microsoft Windows applications. It has similar interfaces to PQDCOM4.DLL and is intended for Use with .NET Applications Built using C# or VB.NET. This library is also free to use and distribute. Its source code is available also on the working group's web site.

PQDiffractor

PQDiffractor is a free software application for viewing, browsing, diagnosing, and converting IEEE 1159.3 PQDIF files and IEEE C37.111 COMTRADE files. It can be used to create interactive charts from PQDIF observations with channels of any quantity type including value logs, waveforms, phasors, mag-dur-time, mag-dur, response, X-Y, X-Y-Z, flash density, cumulative probability frequency, histogram, and 3D histogram. It can convert PQDIF files from native binary PQD to XML. This allows you to read or browse a PQDIF file using a text or XML parser. This feature can be automated using a command line option. PQDiffractor also can determine numerous IEEE Std 1159.3 PQDIF compliance issues.

REVISION OF IEEE 1159.3 PQDIF

IEEE Std 1159.3-2003 was reaffirmed in 2008, which makes it a valid IEEE standard until 2018. Since then, the field of power quality monitoring has evolved. Therefore, the Transmission & Distribution Committee of the IEEE Power & Energy Society is sponsoring a revision of IEEE 1159.3. The revision was approved by the New Standards Committee (NesCom) of the IEEE Standards Association on 21 August 2014. The revision of IEEE 1159.3 has been assigned to the IEEE Working Group for Transfer of Power Quality Data.

This revision effort will update the existing document using the experience gained in twelve years of PQDIF applications, correcting errors, clarifying known points of confusion, and augmenting its list of IDs and values used in the modeling of power quality monitoring instruments and data.

The following changes or additions as part of the IEEE Std 1159.3 revision:

1. Complete editorial changes and corrections to the 2003 edition of IEEE Std. 1159.3
2. Add new ID values for existing PQDIF tags
3. Add new tags and ID values
4. Add new quantity types
5. Add an annex on the representation of PQDIF in XML
6. Add an annex on PQDIF and its relationship to IEC 61850
7. Add an annex on PQDIF and its relationship to IEEE C37.111 COMTRADE

Editorial Changes

The IEEE P1159.3 Working Group plans to complete an editorial review to correct grammatical and typographical errors. Additionally, technical improvements will be completed. For illustration, the revision will include examples of how to use the quantity type known as ID_QT_MAGDURTIME to store the following types of disturbance events lists:

1. rms variation event lists, for storing voltage sags (i.e., voltage dips), voltage swells, and interruptions
2. transient events lists
3. rapid voltage event lists

The last phenomenon in this list is described in detail in Edition 3 of IEC 61000-4-30.

Adding New IDs and New Tags

The working group has created a web site at www.pqdif.info that provides PQDIF users with an online method to submit new IDs for existing tags. The IDs that are submitted online are approved or rejected when the IEEE PQDIF Working Group meets in person, which takes place twice per year.

The IEEE P1159.3 Working Group has approved four new values for tagPhaseID:

- ID_PHASE_LL_MAX: The value representing maximum of 3 line-line values
- ID_PHASE_LL_MIN: The value representing minimum of 3 line-line values
- ID_PHASE_LN_MAX: The value representing maximum of 3 line-neutral values
- ID_PHASE_LN_MIN: The value representing minimum of 3 line-neutral values

The working group has approved the following new values for tagQuantityCharacteristicID:

- ID_QC_REL_HUMIDITY: Relative Humidity, or the ratio of the partial pressure of water vapor in the air–water mixture to the saturated vapor pressure of water at those conditions
- ID_QC_THD_TRIPLEN: Harmonic distortion for harmonics that are a multiple of three
- ID_QC_RVC_DELTA_USS: The absolute difference between the final arithmetic mean 100/120 Urms(1/2) value just prior to the RVC event and the first arithmetic mean 100/120 Urms(1/2) value after the RVC event
- ID_QC_RVC_DELTA_UMAX: The maximum absolute difference between any of the Urms(1/2) values during the RVC event and the final arithmetic mean 100/120 Urms(1/2) value just prior to the RVC event

The quantity characteristics ID_QC_RVC_DELTA_USS and ID_QC_RVC_DELTA_UMAX are added to support the rapid voltage change definitions in the final draft of IEC 61000-4-30 Ed.3.0.

Also, the IEEE P1159.3 working group has approved ID_SERIES_VALUE_TYPE_RMS as a new value for tagSeriesValueTypeID. IEEE 1159.3-2003 included a value type for average, but not for rms, which is needed to properly describe the 10/12 cycle rms average values that are required by IEC 61000-4-30.

Additionally, there have been many new values approved for tagVendorID and tagEquipmentID, which reflect the many new companies and instruments available for power quality monitors.

Representing PQDIF Files in XML

The revision for IEEE 1159.3 will include an annex on how to represent PQDIF files in XML format. The hierarchical, tag-based, extensible structure of XML is a good match for the structure of PQDIF. The task of defining a XML representation of PQDIF is essentially that of defining the XML representation of the PQDIF physical structure. The logical structure of our XML representation is identical to that of the binary PQDIF file. More information is available in [4].

Figure 1 illustrates a typical rms voltage variation due to a power system fault. An abbreviated listing of the XML representation of the measurement follows Figure 1 listed. Each record could contain collections for many channels of data. For clarity, only a single voltage channel (phase C rms voltage) is shown. The listing contains the data source, monitor settings, and observation records for the measurement. As with PQDIF files, the data source record precedes the monitor settings record that precedes the observation record. This order defines the relationship between the records.

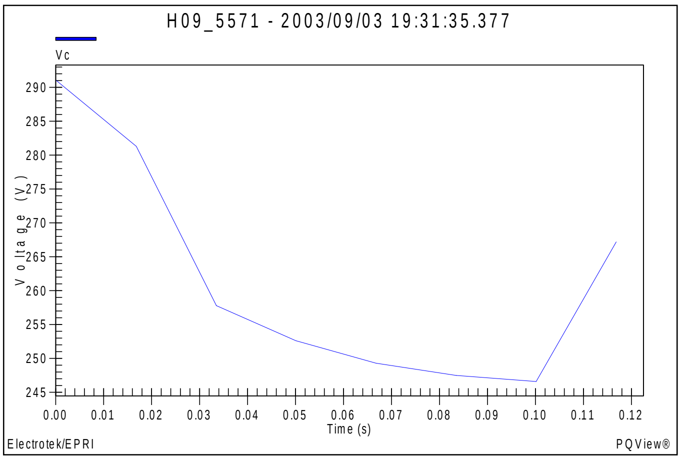

Figure 1: The phase C rms voltage measured during a short duration rms voltage variation


```
<?xml version="1.0"?>
<Records>
 <tagRecDataSource>
 <tagDataSourceTypeID VT=“GUID”>ID_DS_TYPE_MEASURE</tagDataSourceTypeID>
 <tagVendorID VT=“GUID”>ID_VENDOR_BMI</tagVendorID>
 <tagEffective VT=“DT”>2003-06-27 18:50:59.000</tagEffective>
 <tagSerialNumberDS VT=“CHAR1”>71000000</tagSerialNumberDS>
 <tagNameDS VT=“CHAR1" Size="9">H09_5571</tagNameDS>
 <tagChannelDefns>
 <tagOneChannelDefn>
 <tagQuantityMeasuredID VT=“UINT4”>ID_QM_VOLTAGE</tagQuantityMeasuredID>
 <tagChannelName VT=“CHAR1”>Voltage C</tagChannelName>
 <tagPhaseID VT=“UINT4”>ID_PHASE_CN</tagPhaseID>
 <tagQuantityTypeID VT=“GUID”>ID_QT_PHASOR</tagQuantityTypeID>
 <tagSeriesDefns>
 <tagOneSeriesDefn>
 <tagQuantityUnitsID VT=“UINT4”>ID_QU_SECONDS</tagQuantityUnitsID>
 <tagValueTypeID VT=“GUID”>ID_SERIES_VALUE_TYPE_TIME</tagValueTypeID>
 <tagQuantityCharacteristicID VT=“GUID”>ID_QC_RMS</tagQuantityCharacteristicID>
 <tagStorageMethodID VT=“UINT4”>6</tagStorageMethodID>
 <tagSeriesNominalQuantity VT=“REAL8”>-1</tagSeriesNominalQuantity>
 </tagOneSeriesDefn>
 <tagOneSeriesDefn>
 <tagQuantityUnitsID VT=“UINT4”>ID_QU_VOLTS</tagQuantityUnitsID>
 <tagValueTypeID VT=“GUID”>ID_SERIES_VALUE_TYPE_VAL</tagValueTypeID>
 <tagQuantityCharacteristicID VT=“GUID”>ID_QC_RMS</tagQuantityCharacteristicID>
 <tagStorageMethodID VT=“UINT4”>3</tagStorageMethodID>
 <tagSeriesNominalQuantity VT=“REAL8”>277</tagSeriesNominalQuantity>
 </tagOneSeriesDefn>
 </tagSeriesDefns>
 </tagOneChannelDefn>
 </tagChannelDefns>
```

```
 </tagRecDataSource>
 <tagRecMonitorSettings>
 <tagChannelSettingsArray>
 <tagOneChannelSetting>
 <tagTriggerLow VT="REAL8">249.3</tagTriggerLow>
 <tagTriggerHigh VT="REAL8">304.8</tagTriggerHigh>
 <tagNoiseFloor VT="REAL8">5</tagNoiseFloor>
 <tagChannelDefnIdx VT="UINT4">20</tagChannelDefnIdx>
 </tagOneChannelSetting>
 </tagChannelSettingsArray>
 <tagEffective VT="DT">2003-06-27 18:50:59.000</tagEffective>
 </tagRecMonitorSettings>
 <tagRecObservation>
 <tagTimeCreate VT="DT">2003-09-03 19:33:54.000</tagTimeCreate>
 <tagTimeStart VT="DT">2003-09-03 19:31:35.377</tagTimeStart>
 <tagTriggerMethodID
VT="UINT4">ID_TRIGGER_METH_CHANNEL</tagTriggerMethodID>
 <tagTimeTriggered VT="DT">2003-09-03 19:31:35.460</tagTimeTriggered>
 <tagChannelTriggerIdx VT="UINT4">20</tagChannelTriggerIdx>
 <tagChannelInstances>
 <tagOneChannelInst>
 <tagChannelFrequency VT="REAL8">60</tagChannelFrequency>
 <tagChannelDefnIdx VT="UINT4">20</tagChannelDefnIdx>
 <tagSeriesInstances>
 <tagOneSeriesInstance>
 <tagSeriesScale VT="REAL8">0.0166667</tagSeriesScale>
 <tagSeriesOffset VT="REAL8">0</tagSeriesOffset>
 <tagSeriesValues VT="INT2" Size="8">0,1,2,3,4,5,6,7</tagSeriesValues>
 </tagOneSeriesInstance>
 <tagOneSeriesInstance>
 <tagSeriesBaseQuantity VT="REAL8">277</tagSeriesBaseQuantity>
 <tagSeriesScale VT="REAL8">0.1</tagSeriesScale>
 <tagSeriesOffset VT="REAL8">0</tagSeriesOffset>
 <tagSeriesValues VT="INT2"
Size="8">2911,2814,2579,2527,2494,2476,2467,2673</tagSeriesValues>
 </tagOneSeriesInstance>
 </tagSeriesInstances>
 <tagSeriesBaseQuantity VT="REAL8">277</tagSeriesBaseQuantity>
 </tagOneChannelInst>
 </tagChannelInstances>
 </tagRecObservation>
</Records>
```


Standalone PQDIF Observation in XML

Additionally, the working group plans to specify a simpler alternative to the full XML representation of a PQDIF file. This alternative would allow storage of observation records without a data source record or monitor settings record.

Relationship of PQDIF to IEC 61850

IEC 61850-8-1 defines file classes that can be mapped to the Manufacturing Message Specification (MMS) file object, including PQDIF Files and COMTRADE Files. Other supported file types: BIN, DTD, GIF, HTM, TXT, XML, XSD, and ZIP. IEC Technical Committee 57 and IEC Technical Committee 85 are collaborating on a new technical report called IEC TR 61850-90-17. The recommendations of the report are to have a new logical node with the name "Continuous Power Quality recorder" that would be implemented in IEC 61850-7-4 Ed. 3, with IEEE PQDIF as the preferred file format.

The IEEE P1159.3 working group will include an annex that will describe role of PQDIF in the existing IEC 61850 and in the proposed revision.

Relationship of PQDIF to COMTRADE

COMTRADE is an acronym for Common Format for Transient Data Exchange [5]. It is a file format specified in IEEE Std C37.111 for storing oscillography and status data related to power system disturbances. COMTRADE is maintained by the IEEE Relay Communications Subcommittee of the IEEE Power System Relay Committee (PSRC). The 2013 revision of the COMTRADE standard is an IEEE/IEC Dual Logo standard.

The IEEE P1159.3 working group will include an annex that will highlight the similarities and differences of PQDIF and COMTRADE, leveraging the research by the authors in [6].

CONCLUSIONS

IEEE Std 1159.3-2003 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 through its revision 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.

REFERENCES

[1] IEC, 2008, IEC 61000-4-30 ed. 2.0, Electromagnetic Compatibility - Power Quality Measurement Methods.

[2] IEEE, 2008, IEEE Std 1159.3 Recommended Practice for the Transfer of Power Quality Data.

[3] IEEE, 1995, IEEE Std 1159 Recommended Practices on Monitoring Electric Power Quality.

[4] W.W. Dabbs, D.D. Sabin, 2004, "Representation of IEEE Std 1159.3-2002 PQDIF in extensible markup language (XML)," IEEE Power Engineering Society General Meeting, 510-515.

[5] IEEE, 2013, IEEE Std C37.111 IEEE/IEC Measuring Relays and Protection Equipment – Part 24: Common Format for Transient Data Exchange (COMTRADE) for Power Systems.

[6] J.A. King, E.W. Gunther, 2006, "COMTRADE / PQDIF Conversion," Proceedings of the IEEE PES Transmission and Distribution Conference and Exhibition, 359-364

[7] E.W. Gunther, 2006, "On Creating a New Format for Power Quality and Quantity Data Interchange," IEEE PES Transmission and Distribution Conference and Exhibition, 354-358.