My comments about the ADIF standard for HAM Radio logbooks.

Claude Frantz (DJ0OT)

Tentative issue from August 5, 2013

This paper reflects my own opinion about the standard. When not explicitly mentioned, my comments refer to the version 2.2.7 of the ADIF standard which is the last one of the previous main release line, at the time of redaction.[1] The purpose of this paper is deliberately constructive.

1 Introduction

1.1 Benefits

Logbooks according to the ADIF standard are intended for data exchange. Therefore the maximum exchange ability is a major concern. All hardware, operation system or other dependencies have to be avoided. When any standard is used, it has to be freely available and widely accepted. The ADIF standard is an excellent solution to these requirements. It can be used on almost any computer, even very old ones. The computer technology has evolved. Today the XML format is considered to be the more modern and desirable way. There is no major problem to convert ADIF into a XML structure and back. When in the XML format, the data can be processed with all the tools available to manipulate XML data. Dave (AA6YQ) has proposed the ADIX format which is not really more than a XML version of the ADIF format. It should be ensured that the same data can be able to be represented in both formats. Therefore, the same data should be convertible in both directions without data loss.

1.2 Drawbacks

There are many problems related to the ADIF standard. Many ones have been solved but many other ones still remain open. There is a very urgent need for clarification of the existing standard. Many problems encountered in the practice are related to the fact that different implementers do understand the standard in a different manner, because the standard is not explicit enough.

---

1. http://www.adif.org/adif227.html

1.2.1 The America specific view on the world

Some problems are related to the America specific view on the world, ignoring that the world is not limited to the USA and Canada. In fact, many of these problems have been solved in the new versions of the standard, but some other ones remains open. The America centered view to the 'state' has became the 'Primary Administrative Subdivision' and the 'county' has became the 'Secondary Administrative Subdivision', which is a more general view now applicable to most of the countries in the world. The name of the fields did not change, only the meaning. This may be disconcerting for an observer if (s)he does not known about the evolution of the standard. Unfortunately, the '`<ARRL_SECT:>`' is still America (and Canada) specific.

Another problem related to the America specific view on the world is the character encoding. As long as english language is concerned, the basic ASCII code is often sufficient. The only occurring letters are the 26 characters of the latin alphabet. Unfortunately, accented letters are common in many other languages. Such letters can occur on the operator's name or on the QTH. The current ADIF standard still specify that the acceptable characters are those 'whose code lies in the range of 32 through 126, inclusive'. Therefore there is no provision for accented characters. In the proposed ADIX format, the character encoding can be specified because there is a XML structure.

1.2.2 Insufficient specification

ADIF is an exchange format, therefore any aspect of the data should be specified so that nothing remain open about the interpretation of the data. This is an ideal goal which can probably never really be reached. Nevertheless, the specification should try to reach it. In the current version, we are far away from this goal. The result is not surprising: Every one concerned with data in the ADIF format has his/her own idea about the numerous things which are not (or not enough) specified in the standard. This fact result in many incompatibilities in the data exchange. This is exactly the situation which the standard should avoid.

On some occasions, the standard is really explicit. E.g., when referring to the '`<OPERATOR:>`' field: *'if STATION_CALLSIGN is absent, OPERATOR shall be treated as both the logging station's callsign and the logging operator's callsign'*. Unfortunately, important clarifications are missing at many other places.

1.2.3 A surprising development

The new proposed standard 3.0.2 renames ADIX to ADX and includes it in the standard. A schema has been issued for this XML structure, the file is named *adx302.xsd* in this corresponding release. The surprise is that a more precise specification exists now, because it's included in this schema file. Up to now, it's not clear how obligatory this schema will be.

2 My proposals

2.1 My proposal related to the upgrade procedure of the standard

Numerous things are constantly changing which result only in a change in the enumerations contained in the ADIF standard. There are additional modes or countries, etc. Unfortunately a developer of ADIF related software has to read the whole modified standard in order to appreciate what has been changed.

My proposal is to change the release numeration format, so that there will be a part related to all the enumerations and another part related to the remaining matters.

2.2 My proposal related to the file structure

2.2.1 Null fields

In the data processing science, there is a difference between a field containing nothing and an absent field. In the first case, we say: *I known about this matter and what I known is that there is nothing.* In the second case, we say: *I do not known anything about this matter.* Related to the ADIF standard, we can have an absent '<NAME:>' field, which means: *I do not known about the name of the operator of the logged station.* In contrast, an empty '<NAME:0>' field means: *I known about the name of the operator. I known that (s)he has no name.* It is very probable that this statement is wrong.

In order to avoid such problems, the related specification of the standard should explicitly say, that a null field should only be used whey there is really a statement about a matter. Therefore, '<RST:0>' is a nonsense. Further, we have to note that the parsing of ADIF records becomes more simple, if my proposal apply. Instead of programming *if the field exists and if its length is greater than zero and if its contents is valid, then...* can be replaced by the more simple *if the field exists and if its contents is valid, then....*

2.2.2 Padding of fields

Character strings in fields should not be padded with space characters at the begin and/or at the end. Numbers should not be padded with zeroes at the left. Numbers should not be padded with zeroes at the right in the decimal part. Exceptions are allowed when it is a common practice to do so. E.g. it is better to write '<FREQ:5>3.630' rather than '<FREQ:4>3.63'.

2.2.3 Maximum data field length

Some problems can be expected while processing the data because the standard does not set a maximum length for fields. In fact, the definition of some fields imply a specific length. My proposal is the include: *The length of a data field can never be greater than 255 characters.*

2.2.4 Multiple occurrences of the same field in a record

The standard does not explicitly say what is to do when a field having the same field name appears more than one time in the same record. My proposal is the include: *No software emitting data according to the standard should include many times a field with the same field name in a given record. If, while reading data from an ADIF file, a field having the same field name is encountered many times in the same record, only the value set by the last occurrence sets the valid value for the given record.* This definition has been preferred because it's more simple to parse.

2.2.5 Splitting of data fields

It should be disputed to include this limitation in the standard: *A data field, including its field name, length, data type indicator and angle brackets has to be contained in an unique record of the containing file. Such a record can contain many data fields, but no data field can be split over many such records.* This limitation would probably made the parsing easier.

2.2.6 The minimum QSO record

A QSO record is considered valid if it contains only the start or the end time without ambiguity. It is considered a good practice to always insert both. The use of only the start or end time should be limited to QSO of very short duration (e.g. in contests).

— A QSO record is only valid when it contains the callsign of the contacted station in '`<CALL:>`'.

— A QSO record is only valid when it contains the time of the QSO start in '`<TIME_ON:>`' or the time of the QSO end in '`<TIME_OFF:>`'. Whenever possible, both fields should be present.

— A QSO record is only valid when it contains the date of the QSO start in '`<QSO_DATE:>`' and the date of the QSO end in '`<QSO_DATE_OFF:>`'. If the start day and the end day are the same, one of these fields can be omitted. When '`<TIME_OFF:>`' is not present, '`<QSO_DATE_OFF:>`' should not be used. When '`<TIME_ON:>`' is not present, '`<QSO_DATE:>`' should not be used.

— A QSO record is only valid when it contains the transmission frequency of the logging station in '`<FREQ:>`' or the used band in '`<BAND:>`'. Whenever possible, both fields should be included. '`<FREQ:>`' has not to be used to indicate the used band. If the exact frequency is not known and if only the band is known, '`<BAND:>`' has to be used. Whenever possible, the exact frequency should be included in the record.

— If the transmission frequency and the reception frequency are not the same and if '`<FREQ:>`' is used, then '`<FREQ_RX:>`' has to be used too.

— If the transmission band and the reception band are not the same and if '`<BAND:>`' is used, then '`<BAND_RX:>`' has to be used too.

— A QSO record is only valid when it contains the used mode in '<MODE:>'.
— Whenever possible, the exchanged reports should be present in '<RST_SENT:>' and '<RST_RCVD:>'. We have to remember that no RST is exchanged in some contests. This practice is probably more honest than to always send 599 as report. Only reports which have really been exchanged on the air can be included in a QSO record.

2.2.7 The SWL report

Although there is a '<SWL:>' field, the implementation of the SWL report is still missing. In the amateur radio world, SWL reports are an important matter. Therefore, they should be present in the ADIF standard. We need a form of entry saying: *The SWL station X is reporting to the station Y about a QSO which the station Y has conducted with the station Z.* In this report, the SWL station X reports about the signal it has heard from the station Y.

2.2.8 Frequency and band

If the frequency is present in '<FREQ:>' and in '<FREQ_RX:>', the band can be deduced from them. Nevertheless, it is considered a good practice to incorporate both the frequency and the band in the record. This usage should be encouraged. If, for a given region, the frequency is outside of the allocated band, this should be considered as an error in the record. We have to remember that both stations can be in different regions. The frequency bands differ between regions. Further, some national and license class specific limitations may apply.

2.2.9 Multiple QSL cards

Sometimes more than one QSL card have been sent for the same QSO. The cards could have been on the road in different manner (via bureau, direct, hand to hand). Various reasons why exist. In the same manner, many entering cards are possible. The standard should define how such cards have to be handled in an ADIF record. If necessary, additional ADIF fields have to be created in order to allow the handling of multiple QSL cards.

2.3 My proposal related to the header

2.3.1 Default fields

Although the ADIF standard define some fields related to the logging operator, there is no feature to define what the value of such fields should be when specific information is absent in the QSO records. Of course, there can be many operators operating a station in a contest, but there is also the really more common situation where the operator, his/her own QTH, his/her own Maidenhead grid location, his/her own operator name, etc. remain always the same for all the QSO records

in the log. At the present time, the only existing possibility is to introduce this information in every QSO record. An feature to define such default values is very desirable.

My proposal is to allow the inclusion in the header of all fields related to the logging station itself and related to its operator. They are then considered as the default value of these fields. When any of these fields are no present in any QSO record, the value found in the header should be used as default value instead.

2.4 My proposal related to the fields of the QSO records

2.4.1 The end of line sequence within data fields

The current standard defines some 'MultilineString', namely '`<ADDRESS:>`', '`<NOTES:>`', '`<QSLMSG:>`' and '`<RIG:>`'. In the case of '`<ADDRESS:>`', the only real function of the line separator sequence is to separate lines which have to be written separately on a postal envelope. This function could probably better be defined as a separator of sub-fields within the '`<ADDRESS:>`' field.

Things are different for the other 'MultilineString' fields. An 'end of line' separator has only a sense when there are lines. Lines have a length and this length is depending on the string contained in the line, on the used font, on the font size and on the font type. The ADIF standard never defines lines because it does not define geometric areas. Therefore 'end of line' sequences are superfluous in ADIF strings. They should be removed from the standard. My proposal is further: *When* '`<NOTES:>`', '`<QSLMSG:>`' *or* '`<RIG:>`' *fields are encountered while reading an ADIF file, the 'end of line' sequence should be suppressed when it occurs at the begin or at the end of the field and it should be replaced by a space character when this sequence occurs elsewhere in the field.*

2.4.2 The '`<ARRL_SECT:>`' field

It would be better to change the definition of this field to '*national club specific subdivision*'. This could be a possible solution to the problem related to the german *DOK* because the *DOK* is not an administrative subdivision. In contrast, the french *département* is an administrative subdivision so that it's now located in the 'STATE' field (Primary Administrative Subdivision).

In the current standard, there is no counterpart of '`<ARRL_SECT:>`' field related to the logging station. A '`<MY_ARRL_SECT:>`' field should be introduced for this purpose.

2.4.3 The '`<MODE:>`' field

Some modes (e.g. OLIVIA or HELL) have sub-modes which cannot be logged using the present ADIF standard. The creation of a '`<SUBMODE:>`' field is encouraged.

On some instances, there are split mode QSO's. They cannot be logged using the current ADIF standard. The creation of the '<MODE_RX:>' field is proposed, which will then include the mode used on the RX at the logging station side. The '<MODE:>' field should then include the transmission mode used at the logging station.

2.4.4 What is a report ?

Reports are found in the '<RST_SENT:>' and in the '<RST_RCVD:>' field. According to my proposal, two options are available, depending on the first character in these fields.

If the first character is a digit between 1 and 5, then the report is considered a RST report. The length of the data field can then only be 2 or 3. Signal strength values in excess of 9 will not be logged especially. Suffixes like X, C, K, A or D will not be used. Aurora modes will be logged as '<PROP_MODE:AUR>' or '<PROP_MODE:AUE>'.

If the first character is the positive or negative sign, then the report is considered as a signal level relative to the noise level. The sign has to be followed by a number indicating the level in decibels. Real numbers are allowed.

No other formats of reports should be considered valid.

2.4.5 '<COMMENT:>' and '<NOTES:>'

The purpose of both fields is the same. The one should be declared deprecated. My proposal is to declare the '<NOTES:>' field deprecated because it is defined as 'MultilineString'. My further proposal is to include:

— Software emitting ADIF should only emit the '<COMMENT:>' field. Not more the '<NOTES:>' field.
— When a record containing a '<NOTES:>' field and no '<COMMENT:>' field is read, this read field should be converted to a '<COMMENT:>' field while removing the 'end of line' sequences as proposed elsewhere.
— When a record containing both a '<NOTES:>' field and a '<COMMENT:>' field is read, the value used should be the concatenated value of the '<COMMENT:>' and of the '<NOTES:>' field, in this order, while removing the 'end of line' sequences as proposed elsewhere. A space character should be inserted between the two concatenated fields.

2.4.6 Clarification of '<CREDIT_SUBMITTED:>' and '<CREDIT_GRANTED:>'

Awards are numerous and the related rules can often be very different. It's probably an unrealistic goal to try to introduce in the standard all the award related data for all the possible awards. But the interest in awards is so different as radio operators are different. ADIF, as the interchange format, should be able to manage award related fields so that the log files can be used to apply for any possible

award. At the present time, some fields exist in the standard for some awards. This is fine but insufficient. Because of the many differences between awards, there should be an *extensible award related feature* for any possible award in ADIF. It's then the job of the award issuer to define the structure of this information in the ADIF log file. Every software processing the log files has to accept the definitions from the award issuers. An extension of the '`<CREDIT_SUBMITTED:>`' and '`<CREDIT_GRANTED:>`' fields will probably be necessary.

The goal:

— Award related information can be stored in records.
— Award related information does not interfere with other information in the record, including information for other awards and information about QSL cards.
— The award issuer has to define the information to store in the specific fields of the records and to define the rules related to them. The structure of this information has to be compatible with the ADIF structure.

2.4.7 Clarification of '`<QSL_RCVD:>`'

Some debate can be found about the 'V' value of the '`<QSL_RCVD:>`' field. I think that this matter is an award related one and that it should be relocated there. At my knowledge, there is no generally valid and accepted QSL validation process. It is always award related. A QSL card may be rejected for the one award but it may be acceptable for another one.

2.4.8 Clarification of '`<QSL_SENT:>`'

There is an urgent need for a clarification. The definition of '`<QSL_SENT:1>N`' should be changed to *'I have not sent a QSL card up to now and I should not send one in the future'*. Possible reasons are: the remote station does not want a QSL card or it cannot receive it, the operator is 'silent key' now, etc.

A missing '`<QSL_SENT:>`' field indicates that the QSL card generating software should decide itself if a card should be sent or not depending on its own local policy algorithm.

'`<EQSL_QSL_SENT:1>N`' and '`<LOTW_QSL_SENT:1>N`' should be handled in a similar manner.

2.4.9 Clarification of '`<QSL_SENT_VIA:>`'

2.4.10 Additional fields

For some contests and awards, it's necessary to known about the gender of the operator. The boolean field '`<YL:1>`', related to the operator of the contacted station should allow to introduce this information in the log. The same information related to the operator of the logging station should be in the '`<MY_YL:1>`' field.

The '`<QRM:2>`' field reports about the QRM as heard at the logging station site. The first character reports about the nature of the QRM and the second one, which is a digit between 0 and 9, reports about the strength (none to very strong respectively). If the digit is zero, then there is no QRM and the nature is indifferent. My proposed QRM natures are:

A: QRM by amateur stations

C: QRM by commercial or military stations

L: local, man made QRM, usually having some bandwidth

M: QRM from many different sources

X: QRM of unknown origin

The '`<QRN:1>`' field reports about the QRN as heard at the logging station site. The unique digit between 0 and 9 reports about the strength (none to very strong respectively).

The '`<QSB:2>`' field reports about the QSB as heard at the logging station site. The first character reports about the frequency of the QSB and the second one, which is a digit between 0 and 9, reports about the depth (none to very deep respectively). If the digit is zero, then there is no QSB and the frequency is indifferent. The frequency should be coded in this proposed manner:

F: fast QSB

G: very fast QSB

M: medium frequency of QSO

S: slow QSB

T: very slow QSB

Probably, the frequency should be defined in a more scientific manner.

2.5 My proposal related to the use of the standard

When entering data into a record, the most specific field available in the standard has always to be used. The name of the operator and the QTH of the station have to be entered in the '`<NAME:>`' and '`<QTH:>`' fields respectively and not in the '`<NOTES:>`' or '`<COMMENT:>`' fields. A QSO in aurora mode should be logged with '`<PROP_MODE:AUR>`' or '`<PROP_MODE:AUE>`' and not with a suffix to the RST field. Chirps and drift should be noted in the '`<COMMENT:>`' field because no more specific field is available to enter this information.

A general rule should be to be indulgent when reading an ADIF and to be pedantic when generating one. Deprecated fields should be converted to fields according to the current standard. Evident errors should be corrected if possible and an error message should be generated. The rules and recommendations for a good practice should always be followed.

When a program is importing an ADIF file, it should discard no field, so that it should be able to export this eventually updated ADIF file without data loss.

The sequence of fields in the record is meaningless because it does not change the contained data.

Contents