

## ADIF Proposals Issue 24, 28 May 2017

ADIF Proposals Issue 24, 28 May 2017 .....	1
Changes from Previous Version .....	2
Status Key.....	2
Awaiting sponsors.....	2
Items which will have a Poll.....	3
Items which have been approved by a Poll .....	3
Items which have been rejected by a Poll .....	3
Items which will not have a Poll .....	3
Items which have been withdrawn.....	3
Items that have been included in proposed 3.0.6 .....	3
Item 11: Add REP DMP Award .....	3
Item 20.1: Add constraints to applicable Number-type fields AGE, CQZ etc.....	4
Item 20.2: Add constraints to fields: ANT_AZ, ANT_EL .....	7
Item 22: Deprecate LOTW_QSL..., EQSL_QSL..., & QSL... fields and replace with a single field that also includes QRZ.COM .....	7
Item 35: Add the ability to express QSL rejected due to silent key, QSL manager not handling, or QSL bureau not handling. ....	9
Item 36: Provide a means of representing “unknown” values for QSL_SENT, EQSL_SENT (etc.) fields.....	10
Item 47: Add JARL awards to Credit enumeration.....	11
Item 49: Add C4FM to modes .....	11
Item 50: Enhance “Introduction” section .....	11
Item 51: Add field names to the Table of Contents.....	12
Item 52: Align DXCC entity names with ARRL’s list.....	13
Item 54: Add YODX HF Contest.....	13
Item 55: Correct Primary Administrative Subdivision for Italy .....	13
Item 56: Update Primary Administrative Subdivision for Chile .....	13
Item 57: Add ARRL EME Contest.....	13
Item 58: Remove FIPS 6-4 US County Link.....	14
Item 59: Add FSQCALL .....	14
Item 60: Add My Antenna field.....	15

Item 61: Add a REGION field to support WAE and CQ entities.....	15
Item 62: Update Primary Administrative Subdivision for Country Code 5 (Aland Is.) and 224 (Finland) .....	16
Item 63: Add a DARC_DOK field.....	16
Item 64: Update list of Alaska "Counties" .....	16
Item 65: Add a list of US Counties .....	18
Item 66: Mark DXCC entities Midway and Kure Islands as deleted.....	18
Item 67: Extend 2190m upper frequency limit to .1378 MHz .....	18
Item 68: Minor corrections and changes.....	19
Item 69: Add JT10 .....	23
Item 70: Make files exported from the ADIF Specification available .....	23
Item 71: Make ADIF test files available.....	35
Item 72: Deprecate SAT_MODE field.....	38
Item 73: Remove Prefixes from Primary Administrative Subdivision table.....	39
Item 74: Add "ADIF_" to the Sponsored Award Enumeration.....	39
Item 75: Add a DXCC column to the ARRL Sections table.....	40

## Changes from Previous Version

Changes from the previous version of this document are shown by **highlighting**, with deleted text shown by **strikethrough**.

### Status Key

<i>Awaiting comments</i>	There is a minimum 1 week comment period following publication in this document.
<i>Ready for inclusion</i>	The comment period has ended; this is an uncontroversial item that will be included in a draft specification without a poll unless there are adverse comments.
<i>Awaiting sponsors</i>	For a proposal needing a poll, at least two members of the ADIF Voting group must support it beforehand.
<i>Ready for poll</i>	The comment period has ended; a poll can now be initiated.
<i>Poll: Approved</i>	There was a majority of votes for inclusion of the item, which will be included a draft specification
<i>Poll: Rejected</i>	There was not a majority of votes for inclusion of the item, which will be taken no further.
<i>Work in progress</i>	Further work is needed before the item can be considered for inclusion in a draft specification.
<i>Withdrawn</i>	No further action.
<i>Included in a.b.c</i>	Included in proposed specification version <i>a.b.c</i>

### Awaiting sponsors

Item 11: Add REP DMP Award

Item 22: Deprecate LOTW\_QSL\_..., EQSL\_QSL\_..., & QSL... fields and replace with a single field that

also includes QRZ.COM

Item 36: Provide a means of representing "unknown" values for QSL\_SENT, EQSL\_SENT (etc.) fields.

### Items which will have a Poll

Item 47: Add JARL awards to Credit enumeration

Item 58: Remove FIPS 6-4 US County Link

Item 73: Remove Prefixes from Primary Administrative Subdivision table

### Items which have been approved by a Poll

Item 20.1: Add constraints to applicable Number-type fields AGE, CQZ etc.

Item 20.2: Add constraints to fields: ANT\_AZ, ANT\_EL

Item 60: Add My Antenna field

Item 61: Add a REGION field to support WAE and CQ entities

Item 63: Add a DARC\_DOK field

### Items which have been rejected by a Poll

None.

### Items which will not have a Poll

Item 49: Add C4FM to modes

Item 50: Enhance "Introduction" section

Item 51: Add field names to the Table of Contents

Item 52: Align DXCC entity names with ARRL's list

Item 54: Add YO DX HF Contest

Item 55: Correct Primary Administrative Subdivision for Italy

Item 56: Update Primary Administrative Subdivision for Chile

Item 57: Add ARRL EME Contest

Item 59: Add FSQCALL

Item 62: Update Primary Administrative Subdivision for Country Code 5 (Aland Is.) and 224 (Finland)

Item 64: Update list of Alaska "Counties"

Item 69: Add T10

Item 74: Add "ADIF\_" to the Sponsored Award Enumeration

Item 75: Add a DXCC column to the ARRL Sections table

### Items which have been withdrawn

Item 35: Add the ability to express QSL rejected due to silent key, QSL manager not handling, or QSL bureau not handling

Item 66: Mark DXCC entities Midway and Kure Islands as deleted

### Items that have been included in proposed 3.0.6

None.

---

## Item 11: Add REP DMP Award

*Status: Awaiting sponsors*

---

---

There are four categories: HF Fixed, HF Mobile/Portable, VHF Fixed, VHF Mobile/Portable. However, since a single QSO cannot receive credit for both HF and VHF variations, it's not necessary to have additional HF/VHF item(s) in the Credit enumeration.

[11.1] Credit enumeration, add:

Credit For	Sponsor	Award	Facet
DMP-FIXED	REP	Diploma of Portuguese Municipalities (DMP)	Fixed
DMP-MOBILEPORTABLE	REP	Diploma of Portuguese Municipalities (DMP)	Mobile / Portable

[11.2] Sponsored Award enumeration, add:

SPONSOR\_ : REP\_

Sponsoring Organization: Rede dos Emissores Portugueses

[11.3] Secondary Administrative Subdivision Enumeration, add:

Secondary Subdivision: Portuguese Municipalities

Country Code: 149

DXCC Entity: Azores

Number of secondary subdivisions: 19

Award: DMP [http://ct1end.netpower.pt/diplomas/dmp\\_2000.pdf](http://ct1end.netpower.pt/diplomas/dmp_2000.pdf)

Subdivision List: DMP List [http://ct1end.netpower.pt/diplomas/dmp\\_2000.pdf](http://ct1end.netpower.pt/diplomas/dmp_2000.pdf)

Country Code: 256

DXCC Entity: Madeira Is

Number of secondary subdivisions: 11

Award: DMP [http://ct1end.netpower.pt/diplomas/dmp\\_2000.pdf](http://ct1end.netpower.pt/diplomas/dmp_2000.pdf)

Subdivision List: DMP List [http://ct1end.netpower.pt/diplomas/dmp\\_2000.pdf](http://ct1end.netpower.pt/diplomas/dmp_2000.pdf)

Country Code: 272

DXCC Entity: Portugal

Number of secondary subdivisions: 278

Award: DMP [http://ct1end.netpower.pt/diplomas/dmp\\_2000.pdf](http://ct1end.netpower.pt/diplomas/dmp_2000.pdf)

Subdivision List: DMP List [http://ct1end.netpower.pt/diplomas/dmp\\_2000.pdf](http://ct1end.netpower.pt/diplomas/dmp_2000.pdf)

Award Sponsor: REP

Sponsor Defined Code Format: <Four-digit municipality code>

Examples: 0113 -->Oliveira de Azemeis Municipality

Ref. message:

<http://groups.yahoo.com/group/adifdev/message/5598>

---

## Item 20.1: Add constraints to applicable Number-type fields AGE, CQZ etc.

Status: Poll: *Approved*

See further below for a detailed list of changes marked in ***bold italics***:

AGE: 0 to 120

---

---

A\_INDEX: 0 to 400  
CQZ and MY\_CQ\_ZONE: 1 to 40, integers only  
DISTANCE: 0+  
ITUZ and MY\_ITU\_ZONE: 1 to 90, integers only.  
K\_INDEX: 0 to 9, integers only  
MAX\_BURSTS: 0+  
NR\_BURSTS: 0+, integers only  
NR\_PINGS: 0+, integers only  
RX\_PWR: > 0  
SFI: 0+, integers only  
SRX: 0+, integers only  
STX: 0+, integers only  
TX\_PWR: > 0  
TEN\_TEN: > 0, integers only

Add to III.A. Data Types

Data Type: Integer

Data Type Indicator: -

Description: a sequence of one or more Digits representing a decimal integer, optionally preceded by a minus sign (ASCII code 45). Leading zeros are allowed.

Add to III.A. Data Types

Data Type: PositiveInteger

Data Type Indicator: -

Description: an unsigned sequence of one or more Digits representing a decimal integer that has a value greater than zero. Leading zeros are allowed.

Change the AGE field:

Description: the contacted station's operator's age in years *in the range 0 to 120 (inclusive)*

Change the A\_INDEX field:

Description: the geomagnetic A index at the time of the QSO *in the range 0 to 400 (inclusive)*

Change the CQZ field:

Data Type: **PositiveInteger**

Description: the contacted station's CQ Zone *in the range 1 to 40 (inclusive)*

Change the MY\_CQ\_ZONE field:

Data Type: **PositiveInteger**

Description: the logging station's CQ Zone *in the range 1 to 40 (inclusive)*

Change the DISTANCE field:

Description: the distance between the logging station and the contacted station in kilometers via the specified signal path *with a value greater than or equal to zero*

Change the ITUZ field:

Data Type: **PositiveInteger**

Description: the contacted station's ITU zone *in the range 1 to 90 (inclusive)*

---

Change the MY\_ITU\_ZONE field:

Data Type: **PositiveInteger**

Description: the logging station's ITU zone *in the range 1 to 90 (inclusive)*

Change the K\_INDEX field:

Data Type: **Integer**

Description: the geomagnetic K index at the time of the QSO *in the range 0 to 9 (inclusive)*

Change the MAX\_BURSTS field:

Description: maximum length of meteor scatter bursts heard by the logging station, in seconds *with a value greater than or equal to zero*

Change the NR\_BURSTS field:

Data Type: **Integer**

Description: the number of meteor scatter bursts heard by the logging station *with a value greater than or equal to zero*

Change the NR\_PINGS field:

Data Type: **Integer**

Description: the number of meteor scatter pings heard by the logging station *with a value greater than or equal to zero*

Change the RX\_PWR field:

Description: the contacted station's transmitter power in **Watts** *with a value greater than zero*

Change the TX\_PWR field:

Description: the logging station's power in **Watts** *with a value greater than zero*

Change the SFI field:

Data Type: **Integer**

Description: the solar flux at the time of the QSO *with a value greater than or equal to zero*

Change the SRX field:

Data Type: **Integer**

Description: contest QSO received serial number *with a value greater than or equal to zero*

Change the STX field:

Data Type: **Integer**

Description: contest QSO transmitted serial number *with a value greater than or equal to zero*

Change the TEN\_TEN field:

Data Type: **PositiveInteger**

Description: Ten-Ten number *with a value greater than zero*

Change the UKSMG field:

Data Type: **PositiveInteger**

---

---

Description: the contacted station's UKSMG member number **with a value greater than zero**

Sponsors: G3ZOD, AA6YQ

Ref. message: <http://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6070>  
<https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6714>

---

## Item 20.2: Add constraints to fields: ANT\_AZ, ANT\_EL

Status: Poll: *Approved*

See further below for a detailed list of changes marked in **bold italics**:

1. Add the following to the description for ANT\_AZ: True north is 0 degrees with values increasing in a clockwise direction.
2. Add the following to the description for ANT\_EL: The horizon is 0 degrees with values increasing as the angle moves in an upward direction.

I am also proposing that the values for these fields be constrained as follows:

3. Limit the values for ANT\_AZ to the range 0 to 360 degrees. Values that are outside this range are to be normalized on input to be within this range.
4. Limit the values for ANT\_EL to the range -90 degrees (vertically downwards) to 90 degrees (vertically upwards). Values that are outside this range are to be normalized on input to be within this range.

Change the ANT\_AZ field's description to:

the logging station's antenna azimuth, in degrees **with a value between 0 to 360 (inclusive). Values outside this range are deprecated and must be normalized for export (e.g. 370 is exported as 10).**  
**True north is 0 degrees with values increasing in a clockwise direction.**

Change the ANT\_EL field's description to:

the logging station's antenna elevation, in degrees **with a value between -90 to 90 (inclusive). Values outside this range are deprecated and must be normalized for export (e.g. 100 is exported as 80).**  
**The horizon is 0 degrees with values increasing as the angle moves in an upward direction.**

Sponsors: G3ZOD, AA6YQ

Ref. message: <http://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6070>

---

## Item 22: Deprecate LOTW\_QSL..., EQSL\_QSL..., & QSL... fields and replace with a single field that also includes QRZ.COM

Status: *Awaiting sponsors*

The proposal below supersedes the earlier simpler proposal that involved adding QRZ... fields equivalent to the existing QSL..., LOTW..., and EQSL... fields.

---

---

[1] Section III.C.1.b QSO Fields - deprecations

The following fields become “deprecated” (i.e. import only and must not be exported). Their descriptions will have a comment adding referring to the QSL\_STATUS field: “Deprecated; use QSL\_STATUS instead”:

- EQSL\_QSLRDATE
- EQSL\_QSLSDATE
- EQSL\_QSL\_RCVD
- EQSL\_QSL\_SENT
- QSLRDATE
- QSLSDATE
- QSL\_RCVD
- QSL\_RCVD\_VIA
- QSL\_SENT
- QSL\_SENT\_VIA
- LOTW\_QSLRDATE
- LOTW\_QSLSDATE
- LOTW\_QSL\_RCVD
- LOTW\_QSL\_SENT

[2] Section III. Data Types - additions:

Add a new Data Type with:

Data Type: NameValuePair

Data Type Indicator: -

Description: A name-value pair consisting of "name:value". The names are case-insensitive.

Example:

WEATHER:cloudy

Add a new Data Type with:

Data Type: NameValuePairGroup

Data Type Indicator: -

Description: An associated group of NameValuePair types separated by commas.

Example:

WEATHER:cloudy,TEMPERATURE:22

Add a new Data Type with:

Data Type: NameValuePairGroupList

Data Type Indicator: -

Description: An array of NameValuePairGroup types separated by semicolons. Line breaks (ASCII CR (code 13)) and ASCII LF (code 10)) are permitted and ignored

Example:

WEATHER:cloudy,TEMPERATURE:22;QRM:N,QSB:Y

[3] Section III.C.1.b QSO Fields - additions

Add a new field:

Name: QSL\_STATUS

Data Type: NameValuePairGroupList

Enumeration: -

Description:

Details of incoming and outgoing card and electronic QSLs.

The following are defined for use in NameValuePair data types:

---



---

Name	Value (LITERAL or <i>data type</i> )	Description
METHOD	EQSL	eQSL
	LOTW	ARRL Logbook of the World
	QRZCOM	QRZ.COM
	QSL	Card (paper) QSLs
QSLRDATE	<i>Date data type</i>	QSL received date (only valid if QSL_RCVD is Y, I, or V)
QSL_RCVD	<i>QSL_Rcvd enumeration</i>	QSL received status. Default Value: N
QSLSDATE	<i>Date data type</i>	QSL sent date (only valid if QSL_SENT is Y, Q, or I)
QSL_SENT	<i>QSL_Sent enumeration</i>	QSL sent status. Default Value: N
QSL_RCVD_VIA	<i>QSL_Via enumeration</i>	means by which the QSL was received by the logging station
QSL_SENT_VIA	<i>QSL_Via enumeration</i>	means by which the QSL was sent by the logging station

Each NameValuePairGroup must contain exactly one NameValuePair with the name set to METHOD. Each NameValuePairGroup can contain at most one NameValuePair with a particular Name. E.g. a NameValuePairGroup can omit QSLRDATE or include it once.

There must be at most one occurrence of each value of METHOD within the field. E.g. if METHOD:QSL occurs in one NameValuePairGroup, METHOD:QSL must not occur in another NameValuePairGroup.

Examples of NameValuePairGroups:

```
METHOD:LOTW,QSLRDATE:20130910,QSLSDATE:20130822,QSL_RCVD:Y,QSL_SENT:Y
METHOD:EQSL,QSLRDATE:20130910,QSLSDATE:20130822,QSL_RCVD:Y,QSL_SENT:Y
METHOD:QRZCOM,QSLRDATE:20130910,QSLSDATE:20130822,QSL_RCVD:Y,QSL_SENT:Y
METHOD:CARD,QSL_SENT_VIA:B,QSL_RCVD_VIA:D
```

The above examples can be combined into the field by adding semi-colons to form a NameValuePairGroupList.

#### [4] Section III.B. Enumerations QSL Medium Enumeration

Add the following row to the table:

Medium	Description
QRZCOM	QSO confirmation via QRZ.COM

Notes:

[a] The original wording of the proposal specified that the QSL\_STATUS field would not have the data length used in ADI format. However this aspect is not included above since it would require changing to Section IV.A. ADI File Format [http://adif.org.uk/304/ADIF\\_304.htm#ADI\\_File\\_Format](http://adif.org.uk/304/ADIF_304.htm#ADI_File_Format) to include a specific exception for this one field only and would either cause existing parsers to report spurious errors, miss subsequent fields or possibly even cause them to break.

[b] The names of the new data types have been changed to conform with the existing style of the ADIF Specification.

[c] The version of the proposal above has been greatly extended because it had to cover sufficient detail to allow implementation.

Ref. message <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6150>

---

**Item 35: Add the ability to express QSL rejected due to silent key, QSL manager not handling, or QSL bureau not handling.**  
*Status: Withdrawn*

---

---

In III. B Enumerations add values to the QSL Rcvd Enumeration between the 'R' and 'I' entries:

Status	Meaning	Description
<b>B</b>	invalid bureau	the request for a QSL was rejected because the bureau does not handle requests for the contacted station at the QSO's date and time
<b>M</b>	invalid manager	the request for a QSL was rejected because the QSL manager does not handle requests for the contacted station at the QSO's date and time
<b>S</b>	silent key	the request for a QSL was rejected because the contacted operator is now a silent key

In III. C. 1. b. QSO Fields change the QSLRDATE field's description to (new text in ***italic bold***):

"QSL received date

(only valid if QSL\_RCVD is Y, I, **B, M, S**, or V)  
(V deprecated)"

Ref. messages:

<https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6478>

<https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6497>

<https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6434>

---

### Item 36: Provide a means of representing "unknown" values for QSL\_SENT, EQSL\_SENT (etc.) fields.

Status: **Withdrawn** *Awaiting sponsors*

Withdrawn due to lack of sponsors.

In III. B Enumerations add a new value to 16. QSL Sent Enumeration:

Status: U

Meaning: Undetermined

Description:

- Whether to send a QSL or not is undetermined
- Whether to upload a QSO to the online service is undetermined

The reminder of this item is for information and is not for inclusion in the specification.

It is impractical to change the default value for the \_SENT fields because applications importing an ADIF file would not know whether to choose the "old" or "new" default (remember that ADIF\_VER is optional).

HOWEVER, in mitigation, logging applications can explicitly set the new "U" value if they wish.

This is the original request:

*I suggest the following modification of the enumeration applicable to the QSL\_SENT, EQSL\_SENT and LOTW\_SENT fields:*

---

---

1) A new value "D" should be included, meaning that the default QSL policy applicable to the logging station should apply here. Usually this can be the case when the logging operator cannot decide during the QSO, if a card should be send or not, e.g. during a contest or if no access to a database is available. The same apply to EQSL and LOTW.

2) The default value of the QSL\_SENT, EQSL\_QSL\_SENT and LOTW\_QSL\_SENT fields should be changed from "N" to "D", meaning that in the absence of the field, nothing is known about how to handle the QSL.

After an appropriate software has examined and handled such an "D" value, it can handle or delegate an action according to a local policy, changing the value from "D" to another value in accordance to the taken action. E.g. changing to "R" if the software has decided that an QSL should be send, changing to "Y" if the software has generated a QSL, changing to "N" if the software has decided that no QSL should be send.

In addition, the appropriate "QSLSDATE" and "QSL\_SENT\_VIA" should be created or updated accordingly if applicable.

Ref. message: <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6435>

---

## Item 47: Add JARL awards to Credit enumeration

*Status: Work in progress*

Originally JARL awards were going to be included as follows. However, doubt has been raised and clarification is needed. Until such time as that occurs, this item will be dormant.

Credit For	Sponsor	Award	Facet
JARL_MIXED	JARL	All	Mixed
JARL_BAND	JARL	All	Band
JARL_MODE	JARL	All	Mode
JARL_SATELLITE	JARL	All	Satellite
JARL_QRP	JARL	All	QRP

---

## Item 49: Add C4FM to modes

*Status: Ready for inclusion*

Add the following to III.B 9 Modes:

Mode: C4FM

Submodes: -

Description: C4FM 4-level FSK Technology

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6694>

<https://systemfusion.yaesu.com/what-is-system-fusion/>

---

## Item 50: Enhance "Introduction" section

*Status: Ready for inclusion*

In section "I. Introduction":

1. Add a new section heading that includes the first three paragraphs:  
*I.A. Background*
2. Add a new section heading before the fourth paragraph:  
*I.B. Development*

3. Add a new paragraph immediately under “I.B. Development”:  
*“If you would like to discuss the specification, request a change, or report errors, please join the ADIF Developers Group: <https://groups.yahoo.com/neo/groups/adifdev/info>”*
4. Change the start of the final paragraph’s first sentence to:  
*“The ADIF Developers Group produces the ADIF specification and is open to...”*

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6707>

## Item 51: Add field names to the Table of Contents

*Status: Ready for inclusion*

Add hyperlinked field names to the “Table of Contents” sections “III.C.1.a.” and “III.C.1.b.”. Due to the number of fields, this will probably be a list with multiple fields per line.

Here’s a sample of what it would look like:

C. [Fields](#)

1. [ADIF-defined Fields](#)
  - a. [Header Fields](#)

[ADIF\\_VER](#),  
[CREATED\\_TIMESTAMP](#),  
[PROGRAMID](#), [PROGRAMVERSION](#),  
[USERDEF#](#)
  - b. [QSO Fields](#)

[ADDRESS](#), [ADDRESS\\_INTL](#), [AGE](#), [A\\_INDEX](#), [ANT\\_AZ](#), [ANT\\_EL](#), [ANT\\_PATH](#), [ARRL\\_SECT](#), [AWARD\\_GRANTED](#),  
[AWARD\\_SUBMITTED](#),  
[BAND](#), [BAND\\_RX](#),  
[CALL](#), [CHECK](#), [CLASS](#), [CLUBLOG\\_QSO\\_UPLOAD\\_DATE](#), [CLUBLOG\\_QSO\\_UPLOAD\\_STATUS](#), [CNTY](#), [COMMENT](#),  
[COMMENT\\_INTL](#), [CONT](#), [CONTACTED\\_OP](#), [CONTEST\\_ID](#), [COUNTRY](#), [COUNTRY\\_INTL](#), [CQZ](#), [CREDIT\\_SUBMITTED](#),  
[CREDIT\\_GRANTED](#),  
[DISTANCE](#), [DXCC](#),  
[EMAIL](#), [EQ\\_CALL](#), [EQSL\\_QSLRDATE](#), [EQSL\\_QSLSDATE](#), [EQSL\\_QSL\\_RCVD](#), [EQSL\\_QSL\\_SENT](#),  
[FISTS](#), [FISTS\\_CC](#), [FORCE\\_INIT](#), [FREQ](#), [FREQ\\_RX](#),  
[GRIDSQUARE](#), [GUEST\\_OP](#),  
[HRDLOG\\_QSO\\_UPLOAD\\_DATE](#), [HRDLOG\\_QSO\\_UPLOAD\\_STATUS](#),  
[IOTA](#), [IOTA\\_ISLAND\\_ID](#), [ITUZ](#),  
[K\\_INDEX](#),  
[LAT](#), [LON](#), [LOTW\\_QSLRDATE](#), [LOTW\\_QSLSDATE](#), [LOTW\\_QSL\\_RCVD](#), [LOTW\\_QSL\\_SENT](#),  
[MAX\\_BURSTS](#), [MODE](#), [MS\\_SHOWER](#), [MY\\_CITY](#), [MY\\_CITY\\_INTL](#), [MY\\_CNTY](#), [MY\\_COUNTRY](#), [MY\\_COUNTRY\\_INTL](#),  
[MY\\_CQ\\_ZONE](#), [MY\\_DXCC](#), [MY\\_FISTS](#), [MY\\_GRIDSQUARE](#), [MY\\_IOTA](#), [MY\\_IOTA\\_ISLAND\\_ID](#), [MY\\_ITU\\_ZONE](#), [MY\\_LAT](#),  
[MY\\_LON](#), [MY\\_NAME](#), [MY\\_NAME\\_INTL](#), [MY\\_POSTAL\\_CODE](#), [MY\\_POSTAL\\_CODE\\_INTL](#), [MY\\_RIG](#), [MY\\_RIG\\_INTL](#),  
[MY\\_SIG](#), [MY\\_SIG\\_INTL](#), [MY\\_SIG\\_INFO](#), [MY\\_SIG\\_INFO\\_INTL](#), [MY\\_SOTA\\_REF](#), [MY\\_STATE](#), [MY\\_STREET](#),  
[MY\\_STREET\\_INTL](#), [MY\\_USACA\\_COUNTIES](#), [MY\\_VUCC\\_GRIDS](#),  
[NAME](#), [NAME\\_INTL](#), [NOTES](#), [NOTES\\_INTL](#), [NR\\_BURSTS](#), [NR\\_PINGS](#),  
[OPERATOR](#), [OWNER\\_CALLSIGN](#),  
[PFX](#), [PRECEDENCE](#), [PROP\\_MODE](#), [PUBLIC\\_KEY](#),  
[QRZCOM\\_QSO\\_UPLOAD\\_DATE](#), [QRZCOM\\_QSO\\_UPLOAD\\_STATUS](#), [QSLMSG](#), [QSLMSG\\_INTL](#), [QSLRDATE](#),  
[QSLSDATE](#), [QSL\\_RCVD](#), [QSL\\_RCVD\\_VIA](#), [QSL\\_SENT](#), [QSL\\_SENT\\_VIA](#), [QSL\\_VIA](#), [QSO\\_COMPLETE](#), [QSO\\_DATE](#),  
[QSO\\_DATE\\_OFF](#), [QSO\\_RANDOM](#), [QTH](#), [QTH\\_INTL](#),  
[RIG](#), [RIG\\_INTL](#), [RST\\_RCVD](#), [RST\\_SENT](#), [RX\\_PWR](#),  
[SAT\\_MODE](#), [SAT\\_NAME](#), [SFI](#), [SIG](#), [SIG\\_INTL](#), [SIG\\_INFO](#), [SIG\\_INFO\\_INTL](#), [SILENT\\_KEY](#), [SKCC](#), [SOTA\\_REF](#), [SRX](#),  
[SRX\\_STRING](#), [STATE](#), [STATION\\_CALLSIGN](#), [STX](#), [STX\\_STRING](#), [SUBMODE](#), [SWL](#),  
[TEN\\_TEN](#), [TIME\\_OFF](#), [TIME\\_ON](#), [TX\\_PWR](#),  
[UKSMG](#), [USACA\\_COUNTIES](#),  
[VE\\_PROV](#), [VUCC\\_GRIDS](#),  
[WEB](#)
2. [Application-defined Fields](#)
3. [User-defined Fields](#)

---

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6708>

---

### **Item 52: Align DXCC entity names with ARRL's list**

*Status: Ready for inclusion*

Change the DXCC entity names in section "III.B.7 DXCC Entity Code Enumeration" to uppercase versions of the ones currently specified by the ARRL.

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6710>  
<http://www.arrl.org/country-lists-prefixes>

---

### **Item 54: Add YODX HF Contest**

*Status: Ready for inclusion*

The contest administrator has specified the Cabrillo "Contest:" tag as YOHFDX  
Also N1MM Logger+ uses the acronym YOHFDX for this contest.

Add to section III.B.5 Contest ID Enumeration:

Contest-ID: YOHFDX

Description: YODX HF contest <http://www.yodx.ro/>

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6712>  
<http://www.yodx.ro/>

---

### **Item 55: Correct Primary Administrative Subdivision for Italy**

*Status: Ready for inclusion*

For country code 248 (Italy), change "Forlì Cesena" to "Forlì-Cesena"

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6713>

---

### **Item 56: Update Primary Administrative Subdivision for Chile**

*Status: Ready for inclusion*

For "Enumeration for Country Code 112 (Chile)", add:

Code: XIV

Primary Administrative Subdivision: Los Ríos

Prefix: CE6

Code: XV

Primary Administrative Subdivision: Arica y Parinacota

Prefix: CE1

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6724>

---

### **Item 57: Add ARRL EME Contest**

*Status: Ready for inclusion*

ARRL-EME is the Cabrillo contest-id used by N1MM ref. <http://n1mm.hamdocs.com/tiki->

---

---

[download\\_file.php?fileId=738](#)

Add to section III.B.5 Contest ID Enumeration:

Contest-ID: ARRL-EME

Description: ARRL EME contest <http://www.arrl.org/eme-contest>

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6723>

---

## Item 58: Remove FIPS 6-4 US County Link

*Status: Awaiting comments*

In "III.B.12 Secondary Administrative Subdivision Enumeration", the "FIPS 6-4" link is broken because FIPS 6-4 has been replaced by INCITS 38-2009[R2014] which has to be purchased from ANSI for \$38 :

[https://standards.incits.org/apps/group\\_public/project/details.php?project\\_id=206](https://standards.incits.org/apps/group_public/project/details.php?project_id=206).

I propose that the link is removed altogether and footnotes be added underneath the table:

"Alternative lists of US Counties:

[1] "2010 FIPS Codes for Counties and County Equivalent Entities"

<https://www.census.gov/geo/reference/codes/cou.html> provides lists of Counties by State. Subsequent changes are documented in "Substantial Changes to Counties and County Equivalent Entities: 1970-Present"

<https://www.census.gov/geo/reference/county-changes.html>

[2] "INCITS 31-2009[R2014]: Information technology - Codes for the Identification of Counties and Equivalent Areas of the United States, Puerto Rico, and the Insular Areas"

[https://standards.incits.org/apps/group\\_public/project/details.php?project\\_id=204](https://standards.incits.org/apps/group_public/project/details.php?project_id=204) is available for purchase from ANSI <http://webstore.ansi.org> "

Sponsors: G3ZOD, AA6YQ

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6727>

---

## Item 59: Add FSQCALL

*Status: Ready for inclusion*

A request has been made to add FSQCALL to ADIF. FSQCALL is a protocol sent over the transmission mode FSQ. I think the most suitable change is to add FSQCALL as a submode of MFSK. Specifically:

In section "III.B.9 Mode Enumeration", change the MFSK row to include FSQCALL.

In section "III.B.10 Submode Enumeration", add a new row:

Submode: FSQCALL

Mode: MFSK

Description: FSQCall protocol used with FSQ (Fast Simple QSO) transmission mode.

<http://www.qsl.net/z11bpu/MFSK/FSQweb.htm>

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6729>

<http://www.sigidwiki.com/wiki/FSQ>

<http://www.qsl.net/z11bpu/MFSK/FSQweb.htm>

---

---

## Item 60: Add My Antenna field

Status: Poll: *Approved*

Add to III.C.1.b QSO Fields:

Name: MY\_ANTENNA  
Data Type: String  
Enumeration: -  
Description: the logging station's antenna

Name: MY\_ANTENNA\_INTL  
Data Type: IntlString  
Enumeration: -  
Description: the logging station's antenna

Sponsors: G3ZOD, AA6YQ

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6762>

---

## Item 61: Add a REGION field to support WAE and CQ entities

Status: Poll: *Approved*

Add to III.B. Enumerations:

III.B.n Region Enumeration  
Region Entity Code, DXCC Entity Code, Region, Prefix, Applicability  
None, , Not within a WAE or CQ region that is within a DXCC entity,

IV, 206, ITU Vienna, 4U1V, CQ WAE  
AI, 248, African Italy, IG9, WAE  
SY, 248, Sicily, IT9, CQ WAE  
BI, 259, Bear Island, JW/B, CQ WAE  
SI, 279, Shetland Islands, GM/S, CQ WAE  
KO, 296, Kosovo, Z6, CQ WAE  
ET, 390, European Turkey, TA1, CQ

Add to III.C.1.b QSO Fields:

Name: REGION  
Data Type: Enumeration  
Enumeration: Region  
Description: the contacted station's WAE or CQ entity contained within a DXCC entity.  
- The value *None* indicates that the WAE or CQ entity is the DXCC entity in the DXCC field.  
- Nothing can be inferred from the absence of the REGION field.

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6257>  
<https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6273>  
<https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6280>  
<https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6292>  
<https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6293>

---

---

<https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6798>

<https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6784>

---

## Item 62: Update Primary Administrative Subdivision for Country Code 5 (Aland Is.) and 224 (Finland)

*Status: Work in progress*

Update the table for Enumeration for Country Code 5 (Aland Is.) adding a new column *Deleted* and a new record:

051, Märket, OH0, Y

Update the table for Enumeration for Country Code 224 (Finland) adding a new column *Deleted* and the data provided in:

[https://groups.yahoo.com/neo/groups/adifdev/files/Enumeration for country code 224 \(Finland\) 2017-03-16.csv](https://groups.yahoo.com/neo/groups/adifdev/files/Enumeration%20for%20country%20code%20224%20(Finland)%202017-03-16.csv)

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6793>

<https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6791>

---

## Item 63: Add a DARC\_DOK field

*Status: Poll: Approved*

Add to III.C.1.b QSO Fields:

Name: DARC\_DOK

Data Type: Enumeration

Enumeration:

DOKs listed in

[http://www.darc.de/fileadmin/filemounts/referate/dx/contest/wag/Uploads/Kopie\\_von\\_DOKs\\_bands.xls](http://www.darc.de/fileadmin/filemounts/referate/dx/contest/wag/Uploads/Kopie_von_DOKs_bands.xls)

Special DOKs listed in

[https://www.darc.de/fileadmin/filemounts/gs/qs1/sdok/sdok\\_s.pdf](https://www.darc.de/fileadmin/filemounts/gs/qs1/sdok/sdok_s.pdf)

Description:

the contacted station's DARC DOK (District Location Code)

<http://www.darc.de/der-club/referate/dx/contest/wag/en/service/districtsdoks/>

A DOK comprises letters and numbers, e.g. <DARC\_DOK:3>A01

Sponsors: AA6YQ, G3ZOD

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6808>

---

## Item 64: Update list of Alaska "Counties"

*Status: Work in progress*

Change the table in III.B.12.a. Alaskan Counties to the following, noting that in the specification the "Secondary Administrative Subdivision" columns will appear before the "Name" column:

Name	Secondary Administrative Subdivision	Alaska Judicial District	Deleted
------	--------------------------------------	--------------------------	---------

---



Aleutians East	AK,Aleutians East	Alaska Third Judicial District	
<b>Aleutians Islands</b>	AK,Aleutians Islands	Alaska Third Judicial District	Deleted
Aleutians West	AK,Aleutians West	Alaska Third Judicial District	
Anchorage	AK,Anchorage	Alaska Third Judicial District	
<b>Angoon</b>	AK,Angoon	Alaska First Judicial District	Deleted
<b>Barrow</b>	AK,Barrow	Alaska Second Judicial District	Deleted
Bethel	AK,Bethel	Alaska Fourth Judicial District	
Bristol Bay	AK,Bristol Bay	Alaska Third Judicial District	
<b>Cordova-McCarthy</b>	AK,Cordova-McCarthy	Alaska Third Judicial District	Deleted
Denali	AK,Denali	Alaska Fourth Judicial District	
Dillingham	AK,Dillingham	Alaska Third Judicial District	
<b>Fairbanks</b>	AK,Fairbanks	Alaska Fourth Judicial District	
Fairbanks North Star	AK,Fairbanks North Star	Alaska Fourth Judicial District	
<b>First Judicial District</b>	AK,First Judicial District	Alaska First Judicial District	Deleted
<b>Fourth Judicial District</b>	AK,Fourth Judicial District	Alaska Fourth Judicial District	Deleted
Haines	AK,Haines	Alaska First Judicial District	
Juneau	AK,Juneau	Alaska First Judicial District	
<b>Hoonah-Angoon</b>	AK,Hoonah-Angoon	Alaska First Judicial District	
Kenai Peninsula	AK,Kenai Peninsula	Alaska Third Judicial District	
<b>Kenai-Cook Inlet</b>	AK,Kenai-Cook Inlet	Alaska Third Judicial District	Deleted
<b>Ketchikan</b>	AK,Ketchikan	Alaska First Judicial District	Deleted
Ketchikan Gateway	AK,Ketchikan Gateway	Alaska First Judicial District	
<b>Kobuk</b>	AK,Kobuk	Alaska Second Judicial District	Deleted
Kodiak Island	AK,Kodiak Island	Alaska Third Judicial District	
<b>Kuskokwim</b>	AK,Kuskokwim	Alaska Fourth Judicial District	Deleted
<b>Kusilvak</b>	AK,Kusilvak	Alaska Second Judicial District	
Lake and Peninsula	AK,Lake and Peninsula	Alaska Third Judicial District	
<b>Lynn Canal-Icy Straits</b>	AK,Lynn Canal-Icy Straits	Alaska First Judicial District	Deleted
Matanuska-Susitna	AK,Matanuska-Susitna	Alaska Third Judicial District	
Nome	AK,Nome	Alaska Second Judicial District	
North Slope	AK,North Slope	Alaska Second Judicial District	
Northwest Arctic	AK,Northwest Arctic	Alaska Second Judicial District	
<b>Outer Ketchikan</b>	AK,Outer Ketchikan	Alaska First Judicial District	Deleted
<b>Palmer-Wasilla-Talkeetna</b>	AK,Palmer-Wasilla-Talkeetna	Alaska Third Judicial District	Deleted
<b>Petersburg</b>	AK,Petersburg	Alaska First Judicial District	
Pribilof Islands	AK,Pribilof Islands	Alaska Third Judicial District	
<b>Prince of Wales</b>	AK,Prince of Wales	Alaska First Judicial District	Deleted
<b>Prince of Wales-Hyder</b>	AK,Prince of Wales-Hyder	Alaska First Judicial District	

Prince of Wales-Outer Ketchikan	AK,Prince of Wales-Outer Ketchikan	Alaska First Judicial District	Deleted
Saint Matthew Island	AK,Saint Matthew Island	Alaska Fourth Judicial District	
Second Judicial District	AK,Second Judicial District	Alaska Second Judicial District	Deleted
Seward	AK,Seward	Alaska Third Judicial District	Deleted
Sitka	AK,Sitka	Alaska First Judicial District	
Skagway-Hoonah-Angoon	AK,Skagway-Hoonah-Angoon	Alaska First Judicial District	Deleted
Skagway	AK,Skagway	Alaska First Judicial District	
Skagway-Yakutat	AK,Skagway-Yakuta	Alaska First Judicial District	Deleted
Skagway-Yakutat-Angoon	AK,Skagway-Yakutat-Angoon	Alaska First Judicial District	Deleted
Southeast Fairbanks	AK,Southeast Fairbanks	Alaska Fourth Judicial District	
Third Judicial District	AK,Third Judicial District	Alaska Second Judicial District	Deleted
Upper Yukon	AK,Upper Yukon	Alaska Fourth Judicial District	Deleted
Valdez-Chitina-Whittier	AK,Valdez-Chitina-Whittier	Alaska Third Judicial District	Deleted
Valdez-Cordova	AK,Valdez-Cordova	Alaska Third Judicial District	
Wade Hampton	AK,Wade Hampton	Alaska Second Judicial District	Deleted
Wales-Hyder	AK,Wales-Hyder	Alaska First Judicial District	
Wrangell	AK,Wrangell	Alaska First Judicial District	
Wrangell-Petersburg	AK,Wrangell-Petersburg	Alaska First Judicial District	Deleted
Yakutat	AK,Yakutat	Alaska First Judicial District	
Yukon-Koyukuk	AK,Yukon-Koyukuk	Alaska Fourth Judicial District	

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6824>

## Item 65: Add a list of US Counties

*Status: Work in progress*

t.b.s.

Ref.

## Item 66: Mark DXCC entities Midway and Kure Islands as deleted

*Status: Withdrawn*

## Item 67: Extend 2190m upper frequency limit to .1378 MHz

*Status: Work in progress*

The ARRL have reported that the FCC are in the process of adding the band 135.7 to 137.6 kHz. In view of this, when a start date for the band has been announced, change the upper limit on the 2190m band in the Bands enumeration table from .137 to .1376

Ref. <http://www.arrl.org/news/new-bands-fcc-issues-amateur-radio-service-rules-for-630-meters-and-2-200-meters>

## Item 68: Minor corrections and changes

*Status: Work in progress*

In the Table of Contents, change “Propagation Modes” to “Propagation Mode”.

In the Table of Contents, change “DXCC Entity Code Enumeration” to “DXCC Entity Code”.

In section “I. Introduction”, change “internet” to “Internet”.

In section “III.A. Data Types”, remove the data type indicators for Character, Digit, and IntlCharacter. This is because they are of no practical use; they cannot be used in user-defined or application-defined fields because S, N, and I are used with String, Number, and IntlString.

In section “III.A. Data Types”, remove the data type indicators for AwardList, CreditList, and SponsoredAwardList. This is because the letters could better be used with future data types that are of practical value in user-defined or application-defined fields.

In section “III.A. Data Types”, in the description of “AwardList”, remove the spurious “>” at the end of “a comma-delimited list of members of the Award enumeration>”.

In section “III.A. Data Types”, change the description of “Boolean” to:  
if True, the single ASCII character Y or y  
if False, the single ASCII character N or n

In section “III.A. Data Types”, change the description of “Enumeration” to: “an explicit list of legal case-insensitive values represented in ASCII set forth in set notation, e.g. {A, B, C, D}, or defined in a table”. (Note that enumeration values have always been case-insensitive per the start of section “III.B. Enumerations”: “Enumeration values are case insensitive.”)

In section “III.A. Data Types”, change the description of “IntlCharacter” to:  
“a Unicode character (encoded with UTF-8) excluding line break CR (code 13) and LF (code 10) characters”.

In section “III.A. Data Types”, for the data type Location:

- (1) Change a “sequence of characters” to “a sequence of 11 characters”.
- (2) After the bullet point “DDD is a 3...” add a further bullet point: “There is a single space character in between DDD and MM.MMM”.
- (3) Change “MM.MMM is a 6-Digit minutes specifier, where 0 <= MM.MMM <= 59.999 [use leading zeroes]” to “MM.MMM is an unsigned Number minutes specifier with its decimal point in the third position, where 0 <= MM.MMM <= 59.999 [use leading and trailing zeroes]”.

In section “III.B.2 ARRL Section Enumeration”, for the “NWT” row, remove “(import-only)” and mark it as “Deleted” with a date of 2003-11-01 (ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/5566>) instead (the NWT section can validly be exported at any time for a QSO that occurred before NWT was deleted, so “import-only” is incorrect).

---

In section "III.B.2 ARRL Section Enumeration", replace the "Deleted" column with a "Deleted Date" column and add a "From Date" column. Where there are dates available in the table, move them into the "Deleted Date" and "From Date" columns.

In section "III.B.4 Band Enumeration", remove the commas from frequencies that are higher than 1000 MHz (because this is misleading given that the FREQ and FREQ\_RX fields don't allow thousands separators).

Swap the number and order in the contents and body of the sections "III.B.7 DXCC Entity Code Enumeration" and "III.B.8 Credit Enumeration" so that they are in alphabetical order.

In section "III.B.5 Contest ID Enumeration", a few Contest-ID values are in mixed case; for consistency, convert these to uppercase. (This is allowable because enumeration values are case-insensitive.)

In section "III.B.7 DXCC Entity Code Enumeration" change the column name from "Country Code" to "Entity Code" in line with ARRL terminology.

In section "III.B.8 Credit Enumeration", remove the spurious space in "EWAS\_ SATELLITE".

In section "III.B.9 Mode Enumeration", remove "(to be supplied in a future specification)" from the "Description" column title.

In section "III.B.10 Submode Enumeration", add this for the description of submode PCW: "Coherent CW".

In section "III.B.10 Submode Enumeration", remove "(to be supplied in a future specification)" from the "Description" column title.

At the start of "III.B.11 Primary Administrative Subdivision Enumeration", add an explanation of the notation used in the Prefix column. As part of this, for consistency, change the occurrences of a "/" to indicate alternatives to a "," and remove the space in the "EU1, EU2" entry. The explanation is then:

Notation used in Prefix column:

"," indicates alternatives, e.g. "EU1,EU2" would include calls starting with EU1 and UE2.

"[xyz]" indicates one of the characters in brackets, e.g. "UA6[LMNO]" would include calls starting with UA6L, UA6M, UA6N, & UA6O.

"[x-y]" indicates one of the range of characters in brackets, e.g. "UU[0-8]" would include calls starting with UU0, UU1, UU2, UU3, UU4, UU5, UU6, UU7 and UU8. Another example is that UA1[ABF-JLM] would include calls starting with UA1A, UA1B, UA1F, UA1G, UA1H, UA1I, UA1J, UA1L, & UA1M.

"#" indicates any digit, e.g. "LU#[ABC]" would include LU2BCD.

In "III.B.11 Primary Administrative Subdivision Enumeration" add a "Deleted Date" column. For deleted items, this will contain the date on which the item was deleted, and for other items will be blank.

In "III.B.11 Primary Administrative Subdivision Enumeration" some of the CQ Zone and ITU Zone entries have multiple values separated by forward slash characters e.g. 01/02/04. Change the

---

---

forward slash characters to commas so that these values do not cause spreadsheet programs to interpret them as dates. E.g. change "01/02/04" to "01, 02, 04".

In "III.B.11 Primary Administrative Subdivision Enumeration" some of the Primary Administrative Subdivision column values contain comments in addition to a name, e.g.

Ust'-Ordynsky Autonomous Okrug - for contacts made before 2008-01-01

To make the actual names machine-readable and consistent with other tables in the specification, surround the comments in parentheses, e.g. change the above to:

Ust'-Ordynsky Autonomous Okrug (for contacts made before 2008-01-01)

In "III.B.11 Primary Administrative Subdivision Enumeration" for DXCC 15, there are two KK entries. The first needs the text adding "for contacts made before 2007-01-01" (i.e. it is Deleted).

In "III.B.11 Primary Administrative Subdivision Enumeration" for DXCC 15, there are two KT entries. The second first needs the text adding "for contacts made before 2007-01-01" (i.e. it is Deleted).

In section "III.B.18 QSO Upload Status Enumeration" the first table column should have the title "Status" rather than "Via".

In section "III.B.13 Propagation Enumeration" change the title to "III.B.13 Propagation Mode Enumeration".

In section "III.B.18 QSO Upload Status Enumeration" the first table column should have the title "Status" rather than "Via".

In section "III.C.1.b QSO Fields", change the Enumeration column for "CNTY" to: "Secondary Administrative Subdivision (function of STATE field's value)".

In section "III.C.1.b QSO Fields" in the descriptions of the fields RX\_PWR and TX\_PWR, change "watts" to "Watts".

In section "III.C.1.b QSO Fields" change the data type for the IOTA\_ISLAND\_ID and MY\_IOTA\_ISLAND\_ID to PositiveInteger (ref. [Item 20.1](#)).

In section "III.C.1.b QSO Fields", change the Enumeration column for "MY\_CNTY" to: "Secondary Administrative Subdivision (function of MY\_STATE field's value)".

In section "III.C.1.b QSO Fields" change the data type for the IOTA and MY\_IOTA fields to IOTA and add an IOTA data type to section "III.A. Data Types".

In section "III.C.1.b QSO Fields", change the Enumeration column for "MY\_STATE" to: "Primary Administrative Subdivision (function of MY\_DXCC field's value)".

In section "III.C.1.b QSO Fields", change the Enumeration cell for "PROP\_MODE" to: "Propagation Mode".

In section "III.C.1.b QSO Fields", for field "QSO\_COMPLETE", replace the enumeration literal "{Y, N, NIL, ?}" with "QSO Complete" and add an enumeration to section "III.B. Enumerations":

III.B.n QSO Complete  
Abbreviation, Meaning

---

---

Y, yes  
N, no  
NIL, not heard  
?, uncertain

In section “III.C.1.b QSO Fields”, add to the description for SFI the range of 0 to 300

In section “III.C.1.b QSO Fields”, change the Enumeration column for “STATE” to:  
“Primary Administrative Subdivision (function of DXCC field's value)”.

In section “III.C.1.b QSO Fields”, change the Enumeration column for “SUBMODE” to:  
“Submode (function of MODE field's value)”.

In the contents list, “III.B. Enumerations”>Enumerations”, and “III.C.1.b QSO Fields”, replace the spaces in the enumeration names with underscores. This is to facilitate the use of the names in situation where the syntax of names does not permit spaces.

For consistency with the Secondary Administrative Subdivision tables:

- In sections “III.A. Data Types, change the first column title to “Data Type **Name**”.
- In section “III.C.1.a Header Fields”, change the first column title to “**Field Name**”.
- In section “III.C.1.b QSO Fields”, change the first column title to “**Field Name**”.

In section “IV.A.4. Application-defined Fields”, change the first line to: “In ADI files, the form of an Application-defined QSO Field is”

In section “IV.A.4. Application-defined Fields”, on the end of the paragraph “To facilitate importing, display, and editing by other applications”, add the following:  
“, which can be for any Data type except Digit and Character. If a Data type Indicator is not included, the field contents must conform to the MultilineString data type. Note that the first occurrence of an Application-defined field in a file determines its Data type and subsequent occurrences of the field must not attempt to change that Data type.”.

In section “IV.A.5. User-defined Fields”, change the first part of the first line to: “In ADI files, the nth User-defined QSO Field is defined with...”.

In section “IV.A.5. User-defined Fields”, after “D is the Data type indicator”, add “, which can be for any Data type except Digit and Character”.

In section “IV.A.5. User-defined Fields”, add the missing comma to “e.g. ShoeSize{5:20}”.

In section “IV.B.2. Application-defined Fields”, change the first line to: “In ADX files, an Application-defined QSO Field employs the form”.

In section “IV.B.2. Application-defined Fields”, after “DATATYPEINDICATOR is replaced by the data type indicator” add “, which can be for any Data type except Digit, Character, or IntlCharacter. Note that the first occurrence of an Application-defined field in a file determines its Data type and subsequent occurrences of the field must not attempt to change that Data type.”.

In section “IV.B.3. User-defined Fields”, change the first line to: “In ADX files, the nth User-defined QSO Field is defined in the Header by”

---

In section “IV.B.3. User-defined Fields”, after “DATATYPEINDICATOR is replaced by the data type indicator” add “, which can be for any Data type except Digit, Character, or IntlCharacter”.

In section “IV.B.3. User-defined Fields”, add the missing “{” and “}” characters to:

```
<USERDEF FIELDID="n" TYPE="DATATYPEINDICATOR" ENUM="A,B, ... N"  
RANGE="LOWERBOUND:UPPERBOUND">FIELDNAME</USERDEF>
```

(These were accidentally omitted in ADIF 3.0.5; putting them back makes it consistent with the ADIF convention for writing literal enumerations ({...}) as well as the examples and XML Schema files for all releases that have supported ADX.)

Change broken SOTA links from <http://www.sota.org.uk/RulesAndGuidelines> to <http://sota.org.uk/Joining-In/General-Rules>

There are some errors in the XML Schemas to be corrected:

- The validation of MY\_USACA\_COUNTIES and MY\_USACA\_COUNTIES has the use of “:” and “,” reversed.
- The validation of RIG and my RIG\_INTL is being validated against the String and IntlString data types instead of MultilineString and IntlMultilineString.
- The validation of CREDIT\_SUBMITTED and CREDIT\_GRANTED is not allowing the “Credit For” values introduced at ADIF 3.0.5. Correcting this may make the regular expression too long to be usable; if this proves to be the case, then the level of validation will need to be reduced such that the “Credit For” values are not fully validated but other parts of the structure are. For example, in:  
IOTA,WAS:LOTW&CARD,DXCC:CARD  
“IOTA”, “WAS” and “DXCC” would only be validated to be arbitrary strings that start with a letter and comprise letters, numbers and underscores. However, the commas, colons, ampersands, “CARD”, “LOTW”, and “EQSL” would continue to be validated fully.

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6711>  
<https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6840>  
<https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/5388>

---

## Item 69: Add JT10

*Status: Work in progress*

There is a new mode ~~under development~~ named JT10.

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6853>

---

## Item 70: Make files exported from the ADIF Specification available

*Status: Work in progress*

As part of the release of new ADIF versions, a set of machine-readable files for data types, enumerations, and fields will be generated from the ADIF Specification HTML file and included as part of the release. The files will streamline incorporating ADIF items into software and databases.

**~~Since the files are a different way of representing the information in the specification and do not reflect changes to it, they may be added to ADIF 3.0.5 retrospectively and to subsequent versions.~~**

---

---

File types will be: TSV (.tsv), CSV (.csv), XML (.xml), Excel (.xlsx), and OpenOffice Calc (.ods).

The contents will be a generic and fairly literal representation of the tables with header rows including the titles from a table followed by the rows in the table. This includes the structure in the XML which has a generic schema for the tables with a <header> element and subsequent <record> elements that contain <value> records. Why do this rather than have more “type-specific” elements in the XML? It makes the structure of the XML compatible with the structure of the other files and reduces the amount of effort required to exploit the XML in software because the structure is predictable for any “type” of table.

An XML Schema (.xsd) file for the exported XML files may be added at a future release after experience has been gained using the XML files.

An **early** example of the files for ADIF 3.0.5 is available from:

[http://adif.org.uk/working/adifexports\\_2017\\_04\\_21.zip](http://adif.org.uk/working/adifexports_2017_04_21.zip)

Add the following section to the ADIF Specification:

#### **V.C. Exported Files**

The data types, enumerations, and fields tables are exported to a variety of machine-readable files for use in software development. Details are available in the ADIF Resources document section "Exported Files".

Add the following sections to the ADIF Resources document:

#### **IV. Exported Files**

The ADIF Specification’s data types, enumerations, and fields tables are exported to a variety of machine-readable files for use in software development.

The files comprise CSV, TSV, XML, Excel (.xlsx), and OpenOffice Calc (.ods).

There are separate files for data types, enumerations, and fields. Enumerations are also exported to individual files (i.e. a file per enumeration). Additionally, one of the XML files (all.xml) contains all data types, enumerations, and fields. It is anticipated that this and the other XML files will be particularly useful with XSLT transformations to generate (for example) programming language source files and SQL statements.

Warning: Do not use the CSV and TSV files with spreadsheet software such as Excel and OpenOffice Calc. Instead use the Excel (.xlsx) or OpenOffice Calc (.ods) files. This is because by default, spreadsheet software will change CSV and TSV enumeration values that look like numbers and contain leading zeros and values that look superficially like dates and / or times. E.g. 00123 in a CSV or TSV file will end up as 123 in a spreadsheet, whereas the Excel and OpenOffice Calc files are created with all cells set to be of type text, which stops the spreadsheet software guessing the data types of cells.

#### **IV.A. Directory and File Structure**

The exported files are stored in subdirectories within the directory that contains the source ADIF Specification XHTML file (e.g. for ADIF version 3.0.6 the directory name is 306):



---

exports_{version}.zip	A ZIP file containing all the files and directories listed below where {version} is the ADIF version number without dots, e.g. exports_306.zip
exports	Directory for sub-directories based on file type (e.g. .csv).
exports/csv	Directory for CSV (.csv) files.
exports/csv/datatypes.csv	CSV file containing the data types.
exports/csv/enumerations.csv	CSV file containing the enumerations.
exports/csv/enumerations_{name}.csv	CSV file containing the enumeration with the name {name}.
exports/csv/fields.csv	CSV file containing the fields.
exports/tsv	Directory for TSV (.tsv) files.
exports/tsv/datatypes.tsv	TSV file containing the data types.
exports/tsv/enumerations.tsv	TSV file containing the enumerations.
exports/tsv/enumerations_{name}.tsv	TSV file containing the enumeration with the name {name}.
exports/tsv/fields.tsv	TSV file containing the fields.
exports/xlsx	Directory for Excel (.xlsx) files.
exports/xlsx/datatypes.xlsx	Excel file containing the data types.
exports/xlsx/enumerations.xlsx	Excel file containing the enumerations.
exports/xlsx/enumerations_{name}.xlsx	Excel file containing the enumeration with the name {name}.
exports/xlsx/fields.xlsx	Excel file containing the fields.
exports/ods	Directory for OpenOffice Calc (.ods) files.
exports/ods/datatypes.ods	OpenOffice Calc file containing the data types.
exports/ods/enumerations.ods	OpenOffice Calc file containing the enumerations.
exports/ods/enumerations_{name}.ods	OpenOffice Calc file containing the enumeration with the name {name}.
exports/ods/fields.ods	OpenOffice Calc file containing the fields.
exports/xml	Directory for XML (.xml) files.
exports/xml/all.xml	XML file containing the data types, enumerations, and fields.
exports/xml/datatypes.xml	XML file containing the data types.
exports/xml/enumerations.xml	XML file containing the enumerations.
exports/xml/enumerations_{name}.xml	XML file containing the enumeration with the name {name}.
exports/xml/fields.xml	XML file containing the fields.

The enumeration file names for individual enumerations are lowercase and have spaces replaced by underscores. E.g. the ARRL Sections table is exported as the CSV file enumerations\_arrl\_sections.csv

---

#### IV.B. Encoding in Files

---

---

The CSV, TSV, and XML files are encoded as UTF-8 with a byte order mark (BOM) of 0xEF, 0xBB, 0xBF at the start of the file. These 3 bytes can be ignored but are required for compatibility with Microsoft software.

CSV file values are enclosed by double quotes (") and any double quotes embedded within the value are encoded as a pair of double quotes. E.g.

"This value contains a double quotes "" character"

TSV file values are separated by tab characters. A tab character will never occur within a value.

XML file values are encoded in the **same** way as in the TSV files because these correspond to XML Schema 1.0 data types <https://www.w3.org/TR/xmlschema-2/#built-in-datatypes> with the following exceptions:

- Y (yes) values are encoded as the XML Schema boolean data type <https://www.w3.org/TR/xmlschema-2/#boolean> value "true".
- Date values are encoded as the XML Schema dateTime data type <https://www.w3.org/TR/xmlschema-2/#dateTime> UTC values, e.g. 2017-05-22T00:00:00Z

#### IV.C. Exported Data

The tables in the ADIF Specification are exported with all their columns. Any sequences of whitespaces are replaced by a single space. Formatting is stripped out. Some additional columns not found as such in the specification are included - see IV.D. Additional Columns.

In CSV ,TSV, and spreadsheet files, the first record contains a header and this is followed by values records.

Enumeration names have spaces replaced by underscores but the original case of the names is preserved, e.g. ARRL\_Section

Files other than XML files that contain multiple enumerations have multiple header records. When reading these files, the header records can be identified by looking for the text:

Enumeration Name  
in a record's first value.

In the files containing enumerations, the first value in value records contains the name of the enumeration, e.g. ARRL\_Section

In the ADIF specification, different Primary Administrative Subdivision enumeration tables have different columns. In the exported files, each enumeration has the full set of columns, even though some tables in the specification don't include all of the columns. The columns concerned are: "Oblast #", "CQ Zone", and "ITU Zone", and "Prefix".

The Secondary Administrative Subdivision table is not exported at ADIF version 3.0.5 due to the values being external to the specification document.

Depending on the progress of related ADIF specification proposals, files exported from ADIF version 3.0.6 may contain a subset of Secondary Administrative Subdivision data covering Canada and the USA.

"Deleted" columns will either have blank values or contain "true" in XML files or "Deleted" in other file types.

---

---

In the Bands enumeration, frequencies are exported without the thousands separator commas shown in the specification. E.g. 1,240 is exported as 1240

In the Contest ID enumeration, Contest- ID values are always exported as uppercase, e.g. VIRGINIA QSO PARTY

In the Credit enumeration, the value "EWAS\_ SATELLITE" is exported without the embedded space.

In the DXCC Entity Codes enumeration, the first column is exported with the title "Entity Code".

In the Modes enumeration, the Submodes column values are exported without the spaces after commas. E.g. CHIP64,CHIP128

The Propagation enumeration is exported with the name "Propagation\_Mode". In line with this, the PROP\_MODE field is exported with its Enumeration column set to "Propagation\_Mode".

In the QSL\_Upload\_Status enumeration, the first column title "Via" is exported as the title "Status".

In the CREDIT\_SUBMITTED and CREDIT\_GRANTED fields, the data type is exported as "CreditList,AwardList", where the first data type in the list (CreditList) is current and the second item in the list (AwardList) is import-only (deprecated).

#### **IV.D. Additional Columns**

There are some generated columns that don't exist in the ADIF Specification as such:

"Import-only":

Values will be blank or contain the value "false" in XML files and "Import-only" in other file types if the specification indicates somewhere within a table row that the item the row refers to is import-only (deprecated).

"Comments":

Sometimes the table cells with names in (e.g. data type names) contain additional information along the lines of "xxx (import -only; use yyy instead)". In these cases the text within the parentheses is moved into the "Comments" field.

"ADIF Version" and "ADIF Status":

All tables in all files except for the XML files include these columns, which contain the ADIF Specification version (e.g. 3.0.6) and status (Draft, Proposed, or Released).

"Minimum Value" and "Maximum Value":

The data types and fields files include these columns. They contain the minimum and maximum allowed numeric values for the data type or field or are blank.

Note that this does not include all numeric types and fields because ADIF does not specify the minimum or maximum allowed values for number types as imposed by data types within programming languages.

"Header Field":

The ADIF fields files contain this, which is "Y" for ADIF header fields and blank for ADIF record fields.

"DXCC Entity Code":

All the Primary Administrative Subdivision tables in the specification are combined and this column is exported to differentiate between them.

"Contained Within":

---

---

Some of the Primary Administrative Subdivision tables include rows that span all columns in the table and contain the name/details of a locality that encloses the Primary Administrative Subdivisions defined within the following records. These enclosing names/details are exported in this column.

"Deleted":

Any entries within the Primary Administrative Subdivision tables that include the text "for contacts made before" will cause the Deleted column to contain "true" (XML files) or "Deleted" (other file types).

## IV.E. File Structure

### IV.E.1. XML File Structure

The all.xml file has this structure:

```
<?xml version="1.0" encoding="utf-8"?>
<adif version="{version}" status="{status}" created="{date}">
  <dataTypes>
    <header>
      <value>Data Type Name</value>
      <value>{data type title 2}</value>
      ...
    </header>
    <record>
      <value name="Data Type Name">{data type name 1}</value>
      <value name="{data type title 2}">{data type value 1.2}</value>
      ...
    </record>
    <record>
      <value name="Data Type Name">{data type name 2}</value>
      <value name="{data type title 2}">{data type value 2.2}</value>
      ...
    </record>
    ...
  </dataTypes>
  <enumerations>
    <enumeration name="{enumeration name 1}">
      <header>
        <value>Enumeration Name</value>
        <value>{enumeration title 2}</value>
        ...
      </header>
      <record>
        <value name="Enumeration Name">{enumeration name 1}</value>
        <value name="{enumeration title 2}">{enumeration value 1.2}</value>
        ...
      </record>
      <record>
        <value name="Enumeration Name">{enumeration name 1}</value>
        <value name="{enumeration title 2}">{enumeration value 2.2}</value>
        ...
      </record>
      ...
    </enumeration>
    <enumeration name="{enumeration name 2}">
      ...
    </enumeration>
  </enumerations>
  <fields>
    <header>
      <value>Field Name</value>
      <value>{field title 2}</value>
      ...
    </header>
```

---

---

```

<record>
  <value name="Field Name">{field name 1}</value>
  <value name="{field title 2}">{field value 1.2}</value>
  ...
</record>
<record>
  <value name="Field Name">{field name 2}</value>
  <value name="{field title 2}">{field value 2.2}</value>
  ...
</record>
...
</fields>
</adif>

```

where

- {version} is the ADIF version. E.g. 3.0.6
- {status} is Draft, Proposed, or Released
- {date} is the UTC date and time the file was created in XSD date time format, e.g. 2017-04-22T07:42:58Z
- {data type title 2} is the title of the second column in the data types table, etc. E.g. Data Type Indicator
- {data type value 1.2} is the value of the second column of the first row in the data types table, {data type value 2.2} is the value of the second column of the second row in the data types table, etc. E.g. A
- {enumeration name 1} is the name of the first enumeration, {enumeration name 2} is the name of the second enumeration, etc. E.g. Ant\_Path
- {enumeration title 2} is the title of the second column in the first enumeration table, etc. E.g. Abbreviation
- {enumeration value 1.2} is the value of the second column of the first row in the first enumeration table, {enumeration value 2.2} is the value of the second column of the second row in the first enumeration table, etc. E.g. G
- {field title 2} is the title of the second column in the two fields tables combined, etc. E.g. Data Type
- {field value 1.2} is the value of the second column of the first row in the two fields tables combined, {field value 2.2} is the value of the second column of the second row in the two fields tables combined, etc. E.g. ADIF\_VER

Where a value is blank (zero characters in length), the <value name=...> tag will be omitted.

All the XML files have the same overall structure as the all.xml file, the differences being that:

- The datatypes.xml file omits the <enumerations> and <fields> elements.
- The enumerations.xml and named enumeration XML files omit the <dataTypes> and <fields> elements.
- The fields.xml file omits the <dataTypes> and <enumerations> elements.

#### IV.E.2. Spreadsheet File Structure

The Excel and OpenOffice Calc files have the font in header records set to 'bold'. The work sheet names are set as appropriate to:

- "Data Types"
- "Enumerations"
- "{name} Enumeration" truncated to the Excel work sheet name limit of 31 characters.
- "Fields"

where {name} is the enumeration name e.g. "ARRL\_Section Enumeration".

---

The files include some document and custom properties:

Title: "ADIF Specification Export of {item}, Version {version}, Status {status}"  
Author: "ADIF Development Group"  
ADIF Version: "{version}"  
ADIF Status: "{status}"

where

{version} is the ADIF Specification version, e.g. "3.0.6".  
{status} is the ADIF Specification status of "Draft", "Proposed", or "Released".  
{item} is "Data Types", "Enumerations", "Enumeration {name}", or "Fields".  
{name} is an enumeration name, e.g. "ARRL\_Section".

E.g.

Title: "ADIF Specification Export of Enumeration ARRL\_Section, Version 3.0.6,  
Status Proposed"  
Author: "ADIF Development Group"  
ADIF Version: "3.0.6"  
ADIF Status: "Proposed"

#### IV.F. Forwards Compatibility Considerations

Future versions of the ADIF Specification may include changes in the structure of the tables such as:

- a change in a column's title.
- a change in the order of columns.
- additional columns.
- removal of columns.

As far as possible, these types of changes will be avoided, but if they do occur, the files' contents will reflect them.

To cater for this when accessing the CSV, TSV, and spreadsheet files with software, it is recommended that the titles in the header records are used to determine which column is which rather than relying on a column being the "nth" column between successive versions of ADIF. Failing this, software that assumes that the "nth" column has a particular data item in should at least check that the header record contains the expected title for that column.

#### IV.G. Transforming Exported XML using XSLT

While the CSV and TSV files can be read by software to create programming language source files and SQL statements, using XSLT with the exported XML files provides a very convenient alternative.

The following is an example of using XSLT on the fields XML to create C# string constants:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="text" encoding="UTF-8"/>
  <xsl:template match="/">
    <xsl:for-each select="adif/fields/record">
      const string FIELD_<xsl:value-of select="value[@name]"/> = "<xsl:value-of
select="value[@name]"/>";</xsl:for-each>
    </xsl:template>
  </xsl:stylesheet>
```

This is the first part of the output it creates:

```
const string FIELD_ADIF_VER = "ADIF_VER";
const string FIELD_CREATED_TIMESTAMP = "CREATED_TIMESTAMP";
```

---

---

```
const string FIELD_PROGRAMID = "PROGRAMID";
const string FIELD_PROGRAMVERSION = "PROGRAMVERSION";
const string FIELD_USERDEFn = "USERDEFn";
const string FIELD_ADDRESS = "ADDRESS";
const string FIELD_ADDRESS_INTL = "ADDRESS_INTL";
const string FIELD_AGE = "AGE";
const string FIELD_A_INDEX = "A_INDEX";
```

To select (for example) only the fields that are not header fields, the <xsl:for-each> element's select attribute can be altered to include only the <record> elements that do not have a <value> element with a name attribute set to "Header Field":

```
select="adif/fields/record[not (value[@name='Header Field'])]"
```

The following example shows how an individual C# class could be created for each field:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="text" encoding="UTF-8"/>
  <xsl:template match="/">
    <xsl:for-each select="adif/fields/record">
      public class FIELD_<xsl:value-of select="value[@name]"/>
      {
        public const string name = "<xsl:value-of select="value[@name]"/>";
        public const bool header = <xsl:choose>
          <xsl:when test="value[@name='Header Field']">true</xsl:when>
          <xsl:otherwise>>false</xsl:otherwise>
        </xsl:choose>;
      }
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

Here is the first part of the output:

```
public class FIELD_ADIF_VER
{
    public const string name = "ADIF_VER";
    public const bool header = true;
}

public class FIELD_CREATED_TIMESTAMP
{
    public const string name = "CREATED_TIMESTAMP";
    public const bool header = true;
}

public class FIELD_PROGRAMID
{
    public const string name = "PROGRAMID";
    public const bool header = true;
}

public class FIELD_PROGRAMVERSION
{
    public const string name = "PROGRAMVERSION";
    public const bool header = true;
}

public class FIELD_USERDEFn
{
    public const string name = "USERDEFn";
    public const bool header = true;
}
```

---

---

```

}

public class FIELD_ADDRESS
{
    public const string name = "ADDRESS";
    public const bool header = false;
}

```

Here is XSLT that will create equivalent SQL INSERT statements for the fields:

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="text" encoding="UTF-8"/>
    <xsl:template match="/">
        <xsl:for-each select="adif/fields/record">
            INSERT fields (fieldname, header) VALUES ('<xsl:value-of
select="value[@name]"/>', <xsl:choose>
                <xsl:when test="value[@name='Header Field']">1</xsl:when>
                <xsl:otherwise>0</xsl:otherwise>
            </xsl:choose>)</xsl:for-each>
        </xsl:template>
    </xsl:stylesheet>

```

Here is the first part of the output:

```

INSERT fields (fieldname, header) VALUES ('ADIF_VER', 1)
INSERT fields (fieldname, header) VALUES ('CREATED_TIMESTAMP', 1)
INSERT fields (fieldname, header) VALUES ('PROGRAMID', 1)
INSERT fields (fieldname, header) VALUES ('PROGRAMVERSION', 1)
INSERT fields (fieldname, header) VALUES ('USERDEFn', 1)
INSERT fields (fieldname, header) VALUES ('ADDRESS', 0)
INSERT fields (fieldname, header) VALUES ('ADDRESS_INTL', 0)
INSERT fields (fieldname, header) VALUES ('AGE', 0)
INSERT fields (fieldname, header) VALUES ('A_INDEX', 0)

```

Finally, here is a slightly more realistic (and lengthier!) example that converts the Modes enumeration into instances of a C# class:

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="text" encoding="UTF-8"/>
    <xsl:template match="/">using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Adif
{
    public class AdifMode
    {
        private string name;
        private string[] submodes;
        private bool importOnly;
        private string description;

        public string Name
        {
            get { return name; }
        }

        public string[] Submodes
        {
            get { return submodes; }
        }
    }
}

```



---

```

    public bool ImportOnly
    {
        get { return importOnly; }
    }

    public string Description
    {
        get { return description; }
    }

    public AdifMode(
        string name,
        string submodes,
        bool importOnly,
        string description)
    {
        this.name = name;
        this.submodes = submodes.Split(new char[] { ',' },
StringSplitOptions.RemoveEmptyEntries);
        this.importOnly = importOnly;
        this.description = description;

        AdifModes.Add(name, this);
    }

    public static Dictionary<string, AdifMode> AdifModes = new
Dictionary<string, AdifMode>();

    public static void Initialize()
    {
        MessageBox.Show("ARDOP is " + AdifModes["ARDOP"].Description);
        MessageBox.Show("CHIP has " +
AdifModes["CHIP"].Submodes.Length.ToString() + " submodes");
        MessageBox.Show("CLO is " + (AdifModes["CLO"].ImportOnly? "" :
"not") + " import-only");
    }
}
}
</xsl:template>
</xsl:stylesheet>

```

Here's a cut-down version of the output:

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Adif
{
    public class AdifMode
    {
        private string name;
        private string[] submodes;
        private bool importOnly;
        private string description;

        public string Name
        {
            get { return name; }
        }

        public string[] Submodes
        {
            get { return submodes; }
        }
    }
}

```

---

```

public bool ImportOnly
{
    get { return importOnly; }
}

public string Description
{
    get { return description; }
}

public AdifMode(
    string name,
    string submodes,
    bool importOnly,
    string description)
{
    this.name = name;
    this.submodes = submodes.Split(new char[] { ',' },
StringSplitOptions.RemoveEmptyEntries);
    this.importOnly = importOnly;
    this.description = description;

    AdifModes.Add(name, this);
}

public static Dictionary<string, AdifMode> AdifModes = new
Dictionary<string, AdifMode>();

public static void Initialize()
{
    new AdifMode("AM", "", false, "");
    new AdifMode("ARDOP", "", false, "Amateur Radio Digital Open
Protocol");
    new AdifMode("ATV", "", false, "");
    new AdifMode("CHIP", "CHIP64,CHIP128", false, "");
    new AdifMode("CLO", "", false, "");
    new AdifMode("CONTESTI", "", false, "");
    new AdifMode("CW", "PCW", false, "");
    new AdifMode("DIGITALVOICE", "", false, "");
    new AdifMode("DOMINO", "DOMINOEX,DOMINOF", false, "");
    // etc. etc.

    MessageBox.Show("ARDOP is " + AdifModes["ARDOP"].Description);
    MessageBox.Show("CHIP has " +
AdifModes["CHIP"].Submodes.Length.ToString() + " submodes");
    MessageBox.Show("CLO is " + (AdifModes["CLO"].Import ReadOnly
? "" : "not") + " import-only");
    }
}
}

```

When the Intialize method is run, three message boxes will be displayed with showing the following:

```

ARDOP is Amateur Radio Digital Open Protocol
CHIP has 2 submodes
CLO is not import-only

```

#### IV.G. Implementation

Generation of the exported files exploits some meta data included in ADIF Specifications from version 3.0.6 onwards:

- A <meta> tag with the name "adifversion" and content of the ADIF version (e.g. "3.0.6").
-

- 
- A <meta> tag with the name "adifstatus" and content of "Draft", "Proposed", or "Released".
  - The data types table has an id attribute of "Data\_Types" .
  - The Primary Administrative Subdivision enumeration tables have an id attribute of "Enumeration\_Primary\_Administrative\_Subdivision\_{DXCC}" where {DXCC} is the DXCC entity code, e.g. for Canada the id is "Enumeration\_Primary\_Administrative\_Subdivision\_1"
  - All other enumeration tables have an id attribute of "Enumeration\_{name}" where {name} is the enumeration's name with spaces replaced by underscores, e.g. "Enumeration\_Ant\_Path"
  - The fields tables have ID attributes of "Field\_Header" and "Field\_QSO".
  - Data types and fields that have minimum and / or maximum values specified in their descriptions have the values surrounded by <span> tags with the tag's title set to "Minimum" or "Maximum", e.g. <span title="Maximum">99999999</span>

The software source and executable files used to create the files are available from the adif.org.uk website and the ADIF Developers Group files area.

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6857>

---

## Item 71: Make ADIF test files available

*Status: Work in progress*

Over the years there have been requests for ADIF files that can be used to test applications' handing of ADIF.

Within a test QSO, the fields need to have compatible values so that applications being tested don't reject them for reasons such as BAND incompatible with FREQ, DXCC incompatible with CALL, CONT incompatible with CALL, QSO\_DATE after the DXCC in a QSO was deleted, etc. Additionally, each QSO needs a minimum set of fields. The most useful / representative minimum set is likely to be:

QSO\_DATE, TIME\_ON, TIME\_OFF, CALL, FREQ or BAND, and MODE.

The simplest approach would be, for each ADIF release, to hand-craft two files (one for ADI and one for ADX) containing a set of QSOs where each individual field in the specification is included in at least one QSO. At ADIF 3.0.5, this represents around 150 QSOs because some QSO fields will necessarily be included in more than one QSO. For example:

- The fields listed in the paragraph above will be in every QSO.
- All the header fields can be included in the header.
- A QSO testing the STATE field needs to have a DXCC for context.

Additional considerations are that:

1. QSO\_DATE / TIME\_ON / CALL / BAND or FREQ in different QSOs need to be such that applications don't reject a QSO due to being a "duplicate".
  2. QSO\_DATE for "Deleted" items (such as deleted DXCC entities) needs to be before the item was marked as "Deleted".
  3. QSO\_DATE for fields and enumerations marked as "Import-only" (deprecated) needs to be before the item was deprecated.
  4. Taking [2] and [3] above further, in some cases QSO\_DATE also needs to be after an item appeared in an ADIF specification and/or became valid. For example, a QSO containing a
-

---

DXCC field cannot have a QSO\_DATE that is before the date that a DXCC entity code was created.

The ~150 QSOs would however only provide a minimal test because the QSOs would not exercise the wide range of enumeration values in the specification. Ideally, there would be a QSO that tests every value of every enumeration, which would require thousands of QSOs. Hand-crafting and maintaining (between ADIF releases) such a set of test QSOs becomes even more difficult and time-consuming because there are 2,869 different enumeration values in ADIF 3.0.5, requiring more than 2,869 QSOs because more than one field can use a single enumeration; for example, both DXCC and my MY\_DXCC use the DXCC Entity Code enumeration, which has 403 values (including 0), requiring 806 QSOs to cover DXCC Entity Code fully (although in this case, a single QSO could test both DXCC and MY\_DXCC in a single QSO).

The potential provision of an XML file containing a representation of the ADIF specification (if “Item 70: Make files exported from the ADIF Specification” is incorporated) makes creating test files more practical than it has been in the past because it enables automation of a significant part of the work required to create test QSOs. This reduces the effort required both initially and ongoing when a new version of the ADIF specification is released.

So, it is proposed that an application is provided that:

- Creates ADI and ADX files.
- For pairs of \_INTL/non-INTL fields, only includes the \_INTL fields in an ADX file and the non-\_INTL field in an ADI file. (While both the \_INTL and non-\_INTL fields are permitted in ADX files and even within a single QSO record in an ADX file, this is not expected usage.)
- Has the ability to generate basic QSO records with dates / times that move forward in time and varying values for CALL, BAND/FREQ, & MODE to avoid “duplicate QSO” and similar problems, and to allow a basic QSO to have fields under test added to it.
- Creates test QSOs by iterating through the header fields, QSO fields, and enumerations listed in the ADIF specification.
- As far as possible, automatically generates compatible fields in a QSO. For example, by allowing a configuration file to specify a DXCC entity code, allows automatic creation of a QSO that contains an appropriate CALL.
- Allows manual configuration where necessary for each field and enumeration value.

Since the “raw” data should be available in the XML file (all.xml) exported from the ADIF specification, an approach that requires least coding effort would be to use XSLT (Extensible Stylesheet Language Transformations) to transform the XML file into test QSO files.

In order to integrate the use of additional software to (for example) generate basic QSOs automatically such that fields and enumeration values under test can be added to a basic QSO, a library of user functions can be provided that can be invoked from an XSLT file.

The XSLT file(/s) can iterate through the all header and record fields in the specification and provide specific transformations appropriate for each field. Where a field is based on an enumeration (e.g. BAND and BAND\_RX), the transformation for that field will recursively iterate through the enumeration. E.g. for BAND\_RX, a test QSO will be generated for every band along with a random FREQ\_RX within that band. An exception to this is the Secondary Administrative Subdivision enumeration table, which is unlike nearly all other enumeration tables in that the values are external to the specification; in this case, some test QSOs will be “hand-crafted” in the XSLT file. (This will also apply to the DARC\_DOK field if it is incorporated in ADIF 3.0.6 or later.)

---

Another situation where some hand-crafting will be required is in the use of the Credit enumeration, where the test QSOs within the file must be consistent. For example, QSOs that have “<CREDIT\_SUBMITTED:4>DXCC” must each have a unique value in their DXCC fields along with a CALL field that is consistent with the DXCC field.

A starting point is to provide a single pair of files (fields.adi and fields.adx) that cover all fields and enumerations as far as is practical.

There may be issues if applications apply in-depth checking of fields that goes beyond the ADIF specification such that the QSO is valid according to the ADIF specification but not according to the application. For example, where Contest ID values appear in QSOs that have values in the QSO\_DATE, TIME\_ON, BAND, MODE, and other fields that do not conform to the rules of the contest concerned. Where possible, this can be mitigated by trying to make the values as realistic as possible e.g. where a CONTEST\_ID value contains an identifiable reference to a band, set the BAND field accordingly. For example, some Contest IDs have “160” in, representing the 160m band. Another mitigating factor in contests is that (probably) a logging application should not really refuse to allow a user to log “wrong” values for a contest since the user could genuinely make a mistake and rejecting such a QSO is more the province of contest-entry applications than logging applications.

Where applications’ checking becomes an issue, which will only become clear after some trials with test files, consideration can be made to providing primary test files (fields.adx and fields.adi) that omit the problematical values and instead the values concerned are tested in separate test files. For example, if contests do turn out to be an issue, separate contest\_id.adi and contest\_id.adx files could be provided and the Contest Id values in fields.adx and fields.adi be restricted to one or two authentic contest QSOs.

The fields.adi and fields.adx files can provide some testing of file format, particularly for ADI files, such as:

- including comments (represented as text between fields in ADI files and XML comments in ADX files).
- having one QSO per record or one field per record.

Additionally, it is proposed that separate files (no\_header.adi & no\_header.adx) with a few QSOs be provided that (as allowed by the ADIF specification) do not include a header. (In an ADI file with no header, the first character in the file must start with a chevron (<) that is a QSO field, and in an ADX file with no header, the <HEADER> element is present but has no children.)

The first version will not cover items that are:

- Import-only (deprecated)
- Deleted

This is because they add a significant complication in that QSO dates in the files would need to correspond to dates before the items became import-only or deprecated, some of which would require a lot of effort trying to track down dates. In mitigation, it is better to have some test file coverage than the present situation of having none at all.

The directory & file structure in “Item 70: Make files exported from the ADIF Specification” can be extended to include:

---

---

tests_{version}.zip	A ZIP file containing all the files and directories listed below where {version} is the ADIF version number without dots, e.g. tests_306.zip
tests	Directory for sub-directories based on file type (e.g. .adi).
tests/adi	Directory for ADI (.adi) files.
tests/adi/fields.adi	ADI file focussing on fields.
tests/adi/no_header.adi	ADI file containing without a header record.
tests/adx	Directory for ADX (.adx) files.
tests/adx/fields.adx	ADX file focussing on fields.
tests/adx/no_header.adx	ADX file without header records.
tests/xslt	Directory for the XSLT files used to generate the test ADI and ADX files.
tests/xslt/fields.xslt	XSLT file that generates the fields test files.
tests/xslt/no_header.xslt	XSLT file that generates the no_header test files.

The source code and executable files required to generate the test files will be made available on the fists.org.uk website and the ADIF Development (ADIFDev) Yahoo group's Files area.

Limitation: applications being tested should **not** normally **not** attempt to upload test QSOs to services such as Club Log, eQSL, and LoTW. Similarly, they should not attempt to verify QSO fields such as upload statuses using online services because the QSOs will be unknown to the online services.

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6868>

---

## Item 72: Deprecate SAT\_MODE field

*Status: Work in progress*

The SAT\_MODE field's purpose is to store the uplink and downlink bands involved in a satellite QSO (PROP\_MODE of SAT). It has been of data type String in all but ADIF 1 (where it was an Enumeration without any defined values).

However, the same information can be stored in BAND and BAND\_RX, or inferred from FREQ and FREQ\_RX, making SAT\_MODE pointless and adding to the possibility of ADIF fields in a QSO record that have conflicting values. Additionally, not publishing any predefined values for SAT\_MODE has made it less than useful.

It is proposed that SAT\_MODE be "deprecated" in ADIF 3.0.6 and onwards, which means that applications must allow it when importing ADIF files but must not export it in ADIF files. Instead, they should incorporate the bands in BAND (uplink) and BAND\_RX (downlink), and/or put frequencies in FREQ (uplink) and FREQ\_RX (downlink).

The description of SAT\_MODE will be changed to indicate the use of BAND and BAND\_RX along with notes about the mapping of older and newer satellite mode values (such as H and A) onto bands.

---

Additionally, the BAND, BAND\_RX, FREQ, and FREQ\_RX descriptions will have notes added to the effect that BAND must be consistent with FREQ if both are present in a record and ditto for BAND\_RX and FREQ\_RX.

Possibly there should also be a note in the PROP\_MODE field indicating that both BAND and BAND\_RX should be logged for SAT (satellite QSOs).

Useful URLs indicating old and new satellite mode bands are as follows and can be used as the basis of providing some additional information in the SAT\_MODE description:

[https://en.wikipedia.org/wiki/Amateur\\_radio\\_satellite#Mode\\_designators](https://en.wikipedia.org/wiki/Amateur_radio_satellite#Mode_designators)

<http://www.pe0sat.vgnet.nl/satellite/sat-information/modes/>

Specific changes t.b.s.

Sponsors: tbs

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/topics/6616>

---

### **Item 73: Remove Prefixes from Primary Administrative Subdivision table**

*Status: Awaiting comments*

Remove the Prefix column from the Primary Administrative Subdivision tables that contain one and provide links to an external location that provides this information.

The rationale for this proposal is that

1. assembling and maintaining the Prefix information consumes significant time
2. ADIF is driven by volunteers, all of whom are developing amateur radio software and thus by definition short on time
3. if the Prefix information is not assiduously maintained, it will become inaccurate, reflecting poorly on ADIF

Sponsors: AA6YQ, W5IFP

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6925>

<https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6924>

---

### **Item 74: Add "ADIF\_" to the Sponsored Award Enumeration**

*Status: Awaiting comments*

So that there is a sponsor that can be used in creating test QSOs in ADIF files and in examples in the ADIF Specification without encroaching on other organizations' namespaces, add "ADIF\_" to the list in section "III.B.19 Sponsored Award Enumeration" with a footnote:

<i>SPONSOR_</i>	<i>Sponsoring Organization</i>
ADIF_	ADIF - ADIF Development Group
...	

(Note: ADIF\_ is used in examples in this specification and in test QSO records only.)

---

---

Change the examples in the ADIF Specification for AWARD\_SUBMITTED and AWARD\_GRANTED fields to:

<CALL:5>AA6YQ

<AWARD\_SUBMITTED:64>ADIF\_CENTURY\_BASIC,ADIF\_CENTURY\_SILVER,ADIF\_SPECTRUM\_100-160m

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6936>

---

### **Item 75: Add a DXCC column to the ARRL Sections table**

*Status: Awaiting comments*

To help clarify where ARRL Sections are located, add a DXCC column to the ARRL Sections table.

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6936>

---