

## ADIF Proposals Issue 56, 16 July 2024

|  |    |
|--|----|
| ADIF Proposals Issue 56, 16 July 2024.....   | 1  |
| Changes from Previous Version .....  | 2  |
| Status Key.....  | 2  |
| Awaiting sponsors.....   | 3  |
| 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| Items which have been withdrawn.....   | 4  |
| Items that have been included in proposed version 3.1.5 .....                                | 4  |
| Item 161: Review modes implemented by fldigi .....   | 5  |
| Item 160: Add New Zealand Regions award to S.A.S. enumeration.....                           | 5  |
| Item 159: QSLMSG-related clarifications.....   | 5  |
| Item 158: Add QSO Fields QSLMSG_RCVD and QSLMSG_RCVD_INTL.....                               | 5  |
| Item 157: Add QSO fields to support the DARC DCL online service .....                        | 5  |
| Item 156: Revise POTA data types and field definitions.....                                  | 6  |
| Item 155: Kosovo Region / Country .....  | 6  |
| Item 154: Revisions to III.B.17 QSL Via Enumeration .....                                    | 6  |
| Item 153: Review naming of Application-defined files.....                                    | 7  |
| Item 152: Extend (MY_)VUCC_GRIDS to 6-character locators .....                               | 7  |
| Item 151: Add QSO field to represent a Morse code key .....                                  | 7  |
| Item 150: Minor changes and corrections #9 .....   | 9  |
| Item 149: Add primary administrative subdivisions for DXCCs 324, 11, 142 & 70.....           | 10 |
| Item 148: Correct the Region enumeration's African Italy entry .....                         | 12 |
| Item 147: Updates to Contest IDs.....  | 12 |
| Item 146: Provide read-only announcements of new proposed & released ADIF Specifications.... | 13 |
| Item 145: Provide an optional register of PROGRAMID values .....                             | 13 |
| Item 144: Add download date/status fields for QRZ.com .....                                  | 13 |
| Item 143: Add fields for Castles on the Air .....  | 14 |
| Item 135: Review the ADIF specification w.r.t. importing unexpected data.....                | 14 |
| Item 129: Mode/Submode support for GMSK, FSK and FSK-W.....                                  | 14 |

|  |    |
|--|----|
| Item 114: Register ADIF media (aka MIME) types .....                                     | 14 |
| Item 112: Minor changes and corrections #5 .....   | 14 |
| Item 107: Add Support for Extended Character Sets.....                                   | 15 |
| Item 88: Minor Changes and Corrections #3 .....  | 17 |
| Item 87: Correct HFSK submode .....  | 18 |
| Item 85: Add Primary Administrative Subdivision for Norway .....                         | 18 |
| Item 83: Remove Digit, Character, and IntlCharacter from Data Types table .....          | 19 |
| Item 82: Add section describing the meaning of Deleted .....                             | 20 |
| Item 81: Replace “Y” in Deleted columns with “Deleted” .....                             | 20 |
| Item 80: Clarifications on the fields BAND, BAND_RX, FREQ, & FREQ_RX .....               | 21 |
| Item 71: Make ADIF test files available.....   | 21 |
| Item 70: Make files exported from the ADIF Specification available .....                 | 25 |
| Item 65: Add a list of US Counties .....   | 36 |
| Item 62.2: Update Primary Administrative Subdivision for Country Code 224 (Finland)..... | 36 |
| Item 58: Remove FIPS 6-4 US County Link.....   | 36 |
| Item 47: Add JARL awards to Credit enumeration.....                                      | 37 |
| Item 11: Add REP DMP Award .....   | 37 |

## 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 154: Revisions to III.B.17 QSL Via Enumeration

## Items which will have a Poll

Item 47: Add JARL awards to Credit enumeration

Item 58: Remove FIPS 6-4 US County Link

Item 80: Clarifications on the fields BAND, BAND\_RX, FREQ, & FREQ\_RX

Item 107: Add Support for Extended Character Sets

Item 129: Mode/Submode support for GMSK, FSK and FSK-W

Item 143: Add fields for Castles on the Air

Item 144: Add download date/status fields for QRZ.com

Item 151: Add QSO field to represent a Morse code key

Item 157: Add QSO fields to support the DARC DCL online service

Item 158: Add QSO Fields QSLMSG\_RCVD and QSLMSG\_RCVD\_INTL

## Items which have been approved by a Poll

None.

## Items which have been rejected by a Poll

None.

## Items which will not have a Poll

Item 62.2: Update Primary Administrative Subdivision for Country Code 224 (Finland)

Item 81: Replace “Y” in Deleted columns with “Deleted”

Item 83: Remove Digit, Character, and IntlCharacter from Data Types table

Item 85: Add Primary Administrative Subdivision for Norway

Item 88: Minor Changes and Corrections #3

Item 97: Add Clarifications on ADIF to Introduction Section

Item 112: Minor changes and corrections #5

Item 114: Register ADIF media (aka MIME) types

Item 135: Review the ADIF specification w.r.t. importing unexpected data

Item 145: Provide an optional register of PROGRAMID values

Item 146: Provide read-only announcements of new proposed & released ADIF Specifications

Item 147: Updates to Add new Contest IDs

Item 148: Correct the Region enumeration’s African Italy entry

Item 149: Add primary administrative subdivisions for DXCCs 324, 11, 142 & 70

Item 150: Minor changes and corrections #9

Item 152: Extend (MY\_)VUCC\_GRIDS to 6-character locators

Item 153: Review naming of Application-defined files

Item 155: Kosovo Region / Country

Item 156: Revise POTA data types and field definitions

Item 159: QSLMSG-related clarifications

Item 160: Add New Zealand Regions award to S.A.S. enumeration

Item 161: Review modes implemented by fldigi

### **Items which have been withdrawn**

None

### **Items that have been included in proposed version 3.1.5**

None.

### Item 161: Review modes implemented by fldigi

Status: *Work in progress*

There are discrepancies between the modes supported by fldigi and ADIF; details have been forwarded to Dave W1HJK for review.

Ref. <https://groups.io/g/adifdev/message/9724>

### Item 160: Add New Zealand Regions award to S.A.S. enumeration

Status: *Awaiting comments*

Secondary Subdivision / Country Code / DXCC Entity / Number of secondary subdivisions / Award / Subdivision List / Award Sponsor / Sponsor Defined Code Format / Examples

NZ Regions / 120 / New Zealand / n / NZR / NZART Awards

<https://www.nzart.org.nz/assets/activity/awards/Awards-NZART-2023.pdf> / NZART / <Region> COMMA <District> / Bay of Plenty, Kawerau

Ref. <https://groups.io/g/adifdev/message/9721>

### Item 159: QSLMSG-related clarifications

Status: *Awaiting comments* ~~*Work in progress*~~

Add a note to COMMENT and COMMENT\_INTL fields:

“For a message for the contacted station's operator to be incorporated in a paper or electronic QSL, use the QSLMSG or QSLMSG\_INTL field”

Ref. <https://groups.io/g/adifdev/message/9426>

Change the description of the QSLMSG and QSLMSG\_INTL fields from:

“QSL card comment”

to

“a message for the contacted station's operator to be incorporated in a paper or electronic QSL”

Ref. <https://groups.io/g/adifdev/message/9426>

### Item 158: Add QSO Fields QSLMSG\_RCVD and QSLMSG\_RCVD\_INTL

Status: *Awaiting comments* ~~*Work in progress*~~

**Field Name / Data Type / Enumeration / Description**

QSLMSG\_RCVD / MultilineString / - / a message addressed to the logging station's operator incorporated in a paper or electronic QSL

QSLMSG\_RCVD\_INTL / IntlMultilineString / - / a message intended for the logging station's operator incorporated in a paper or electronic QSL

Sponsors: G3ZOD, NY4I, AD1C

Ref. <https://groups.io/g/adifdev/message/9357>

### Item 157: Add QSO fields to support the DARC DCL online service

Status: **Awaiting comments** ~~Work in progress~~

Add to QSO fields including links to DCL <https://www.darc.de/en/der-club/referate/committee-dx/diplome/darc-community-logbook-dcl/>

| Field Name   | Data Type   | Enumeration | Description   |
|--------------|-------------|-------------|---|
| DCL_QSLRDATE | Date        |             | date QSL received from DCL<br><br>(only valid if DCL_QSL_RCVD is Y, I, or V)<br>(V import-only) |
| DCL_QSLSDATE | Date        |             | date QSL sent to DCL<br><br>(only valid if DCL_QSL_SENT is Y, Q, or I)                          |
| DCL_QSL_RCVD | Enumeration | QSL Rcvd    | DCL QSL received status<br><br>(V import-only)<br><br>Default Value: N                          |
| DCL_QSL_SENT | Enumeration | QSL Sent    | DCL QSL sent status<br><br>Default Value: N   |

Sponsors: AA6YQ, G3ZOD

Ref. <https://groups.io/g/adifdev/message/9420>

### Item 156: Revise POTA data types and field definitions

Status: *Work in progress*

t.b.s.

Ref.

### Item 155: Kosovo Region / Country

Status: **Awaiting comments** ~~Work in progress~~

In section III.B.20 Region Enumeration, change:

Region Identity Code / DXCC Entity Code / Region / Prefix / Applicability  
KO / 296 / Kosovo / Z6 / CQ, WAE

to:

KO / 296 / Kosovo (for contacts made before 2018-01-21) / Z6 / CQ, WAE

Ref. <https://groups.io/g/adifdev/message/8881>

### Item 154: Revisions to III.B.17 QSL Via Enumeration

Status: **Awaiting sponsors** ~~Work in progress~~

Change values from:

B                      bureau

D direct  
 E electronic  
 M (import-only) manager (import-only)

to:

B bureau  
 C Club Log OQRS  
 D direct  
 F file-based  
 P PayPal "Direct" OQRS  
 E (import-only) electronic (import-only)  
 M (import-only) manager (import-only)

Add a footnote that examples of F / "file-based" are image and PDF files

**NOTE: No sponsors have come forward, so this item will be removed in the next version of this document.**

Ref. <https://groups.io/g/adifdev/message/9334>

### Item 153: Review naming of Application-defined files

Status: **Awaiting comments** **Work in progress**

Include rules for naming Application-defined fields.

Ref. <https://groups.io/g/adifdev/message/9324>

### Item 152: Extend (MY\_)VUCC\_GRIDS to 6-character locators

Status: **Awaiting comments** **Work in progress**

Extend the definition of (MY\_)VUCC\_GRIDS to include 6-character locators to bring them into line with LoTW and include in examples.

Ref. <https://groups.io/g/adifdev/message/8851>

### Item 151: Add QSO field to represent a Morse code key

Status: *Work in progress*

**Add QSO fields:**

| Field Name        | Data Type   | Enumeration    | Description   |
|-------------------|-------------|----------------|---|
| MORSE_KEY_INFO    | String      |                | details of the contacted station's Morse key (e.g. make, model, etc). |
| MORSE_KEY_TYPE    | Enumeration | Morse Key Type | the contacted station's Morse key type (e.g. straight key, bug, etc). |
| MY_MORSE_KEY_INFO | String      |                | details of the logging station's Morse key (e.g. make, model, etc).   |

|                   |             |                |   |
|-------------------|-------------|----------------|---|
| MY_MORSE_KEY_TYPE | Enumeration | Morse Key Type | the logging station's Morse key type (e.g. straight key, bug, etc). |
|-------------------|-------------|----------------|---|

Add enumeration Morse Key Type:

| Abbreviation | Meaning                                 | Characteristics   | Morse Composition  | Examples   |
|--------------|---|---|--|--|
| SK           | Straight Key                            | a single lever which actuates a single switch   | a human makes the dits and dahs and builds characters                      | Lionel J-38  |
| SS           | Sideswiper                              | a single lever which actuates a single switch when moved in one direction and actuates a single switch when moved in another direction. Both switches are wired in parallel.  | a human makes the dits and dahs and builds characters                      | W1SFR<br>Green<br>Machine<br>Torsion Bar<br>Cootie |
| BUG          | Mechanical semi-automatic keyer or Bug  | a single lever which closes a switch when moved in one direction and actuates a spring and pendulum mechanism which opens and closes a switch when moved in another direction. Both switches are wired in parallel. | a machine makes the dits and a human makes the dahs and builds characters. | Vibroplex<br>Blue Racer<br>Deluxe                  |
| FAB          | Mechanical fully-automatic keyer or Bug | a single or dual lever key which closes a switch when moved in either of two directions actuates a spring and pendulum mechanism which opens and closes a switch. Both switches are wired in parallel.              | a machine makes the dits and a human makes the dahs and builds characters. | GHD<br>GN209FA<br>fully-<br>automatic<br>bug       |
| SLP          | Single-Lever Paddle                     | a single lever which closes a switch when moved in one direction and closes a separate switch when moved in another direction.  | a machine makes the dits and the dahs and a human builds the characters.   | American<br>Morse Mini-<br>B                       |
| DLP          | Dual-Lever Paddle                       | two levers which each close an independent switch when moved in one direction.  | a machine makes the dits and the dahs and a human builds the characters.   | Begali<br>Sculpture                                |
| CPU          | Computer Driven                         | an electronic device performs the closing of the switch.  | a machine makes the dits and dahs to build the characters.                 | N1MM+<br>Logging<br>Software                       |



Ref. <https://groups.io/g/adifdev/message/8931>  
<https://groups.io/g/adifdev/message/9443>  
<https://groups.io/g/adifdev/message/9600>

## Item 150: Minor changes and corrections #9

Status: *Awaiting comments* *Work in progress*

Add explanation of Import and Export.  
Insert a new section before "I.C. Applicability":

I.C. Terminology

Export: Saving data from an application or service into ADIF format.

Import: Loading data in ADIF format into an application or service.

Change the first line in the current section "I.C. Applicability" from:

ADIF's purpose is to enable the reliable transfer of amateur radio information from one application or service to another.

to:

ADIF's purpose is to enable the reliable transfer of amateur radio information from one application or service ("export") via an intermediate form (ADIF) to another application or service ("import").

Ref. <https://groups.io/g/adifdev/message/8838>

In section III.B.6 Continent Enumeration, change the name "Oceana" to "Oceania".

Ref. <https://groups.io/g/adifdev/message/9207>

In section III.A.1 Maidenhead Locators, change the examples so that the same locator occurs in all of them. Also add a note:

*"If both GRIDSQUARE and VUCC\_GRIDS are exported in a QSO record, the locator in GRIDSQUARE should be consistent with the locators in VUCC\_GRIDS. Similarly, MY\_GRIDSQUARE should be consistent with MY\_VUCC\_GRIDS.*

*For example, the following is consistent because the locators in VUCC\_GRIDS have four characters and the first four characters in GRIDSQUARE are the same as those in one of the locators in VUCC\_GRIDS:*

*<MY\_GRID\_SQUARE:6>EM97AB <VUCC\_GRIDS:19>EN98,FM08,EM97,FM07"*

In section "IV.A.6. ADI Records" correct "the the".

In the POTARef datatype definition, change the example:

K-10000, 5-digit park numbers are reserved for future use

to

K-10000, Power City State Fish and Wildlife Area

Ref. <https://groups.io/g/adifdev/message/9323>

For clarity in the QSO fields VUCC\_GRIDS, MY\_VUCC\_GRIDS, USACA\_COUNTIES and MY\_USACA\_COUNTIES, add to the start of the description:

"a comma-delimited list of"

Bring the DXCC III.B.8 DXCC Entity Code Enumeration back in line with the ARRL list.

Ref. <https://groups.io/g/adifdev/message/9355>

Corrections/updates to the Primary Administrative Subdivisions.  
<https://www.iso.org/obp/ui/#iso:code:3166:AR> DXCC 100 Argentina  
<https://www.iso.org/obp/ui/#iso:code:3166:BR> DXCC 108 Brazil  
<https://www.iso.org/obp/ui/#iso:code:3166:CL> DXCC 112 Chile  
<https://www.iso.org/obp/ui/#iso:code:3166:DE> DXCC 230 Germany  
<https://www.iso.org/obp/ui/#iso:code:3166:ES>

DXCC 281 Spain  
DXCC 21 Balearic Is.  
DXCC 29 Canary Is.  
DXCC 32 Ceuta & Melilla

<https://www.iso.org/obp/ui/#iso:code:3166:FR> DXCC 227 France  
<https://www.iso.org/obp/ui/#iso:code:3166:IE> DXCC 245 Ireland  
<https://www.iso.org/obp/ui/#iso:code:3166:MX> DXCC 50 Mexico  
<https://www.iso.org/obp/ui/#iso:code:3166:NL> DXCC 263 Netherlands  
<https://www.iso.org/obp/ui/#iso:code:3166:NZ> DXCC 170 New Zealand  
<https://www.iso.org/obp/ui/#iso:code:3166:PG> DXCC 163 Papua New Guinea  
<https://www.iso.org/obp/ui/#iso:code:3166:PH> DXCC 375 Philippines  
<https://www.iso.org/obp/ui/#iso:code:3166:PY> DXCC 132 Paraguay  
<https://www.iso.org/obp/ui/#iso:code:3166:IT> DXCC 225 Sardinia  
<https://www.iso.org/obp/ui/#iso:code:3166:RO> DXCC 275 Romania  
<https://www.iso.org/obp/ui/#iso:code:3166:SE> DXCC 284 Sweden  
<https://www.iso.org/obp/ui/#iso:code:3166:UY> DXCC 144 Uruguay

Ref. <https://groups.io/g/adifdev/message/9355>  
<https://groups.io/g/adifdev/message/9418>

## Item 149: Add primary administrative subdivisions for DXCCs 324, 11, 142 & 70

Status: **Awaiting comments** ~~Work in progress~~

I propose that we add enumerations for Primary Administrative Subdivisions in India (country code 324), Andaman and Nicobar Islands (country code 11), and Lakshadweep Islands (country code 142) to the next version of ADIF (3.1.5) using the names defined by ISO 3166:

I propose that the Primary Administrative Subdivisions for Cuba (DXCC country code 70) be added to ADIF based on the information in... ISO 3166.

### Enumeration for Country Code 11 (Andaman & Nicobar Is.)

| Code | Primary Administrative Subdivision            |
|------|---|
| AN   | Andaman and Nicobar Islands (Union territory) |

### Enumeration for Country Code 70 (Cuba)

| Code | Primary Administrative Subdivision |
|------|------------------------------------|
| 15   | Artemisa (province)                |
| 09   | Camagüey (province)                |
| 08   | Ciego de Ávila (province)          |
| 06   | Cienfuegos (province)              |
| 12   | Granma (province)                  |

|    |  |
|----|--|
| 14 | Guantánamo (province)                      |
| 11 | Holguín (province)                         |
| 99 | Isla de la Juventud (special municipality) |
| 03 | La Habana (province)                       |
| 10 | Las Tunas (province)                       |
| 04 | Matanzas (province)                        |
| 16 | Mayabeque (province)                       |
| 01 | Pinar del Río (province)                   |
| 07 | Sancti Spíritus (province)                 |
| 13 | Santiago de Cuba (province)                |
| 05 | Villa Clara (province)                     |

**Enumeration for Country Code 142 (Lakshadweep Is.)**

| <b>Code</b> | <b>Primary Administrative Subdivision</b> |
|-------------|---|
| LD          | Lakshadweep (Union territory)             |

**Enumeration for Country Code 324 (India)**

| <b>Code</b> | <b>Primary Administrative Subdivision</b>                  |
|-------------|--|
| AP          | Andhra Pradesh (state)                                     |
| AR          | Arunāchal Pradesh (state)                                  |
| AS          | Assam (state)  |
| BR          | Bihār (state)  |
| CH          | Chandīgarh (Union territory)                               |
| CG          | Chhattīsgarh (state)                                       |
| DL          | Delhi (Union territory)                                    |
| DH          | Dādra and Nagar Haveli and Damān and Diu (Union territory) |
| GA          | Goa (state)  |
| GJ          | Gujarāt (state)  |
| HR          | Haryāna (state)  |
| HP          | Himāchal Pradesh (state)                                   |
| JK          | Jammu and Kashmīr (Union territory)                        |
| JH          | Jhārkhand (state)  |
| KA          | Karnāataka (state)   |
| KL          | Kerala (state)   |
| LA          | Ladākh (Union territory)                                   |
| MP          | Madhya Pradesh (state)                                     |
| MH          | Mahārāshtra (state)  |
| MN          | Manipur (state)  |
| ML          | Meghālaya (state)  |
| MZ          | Mizoram (state)  |
| NL          | Nāgāland (state)   |

OD Odisha (state)  
PY Puducherry (Union territory)  
PB Punjab (state)  
RJ Rājasthān (state)  
SK Sikkim (state)  
TN Tamil Nādu (state)  
TS Telangāna (state)  
TR Tripura (state)  
UP Uttar Pradesh (state)  
UK Uttarākhand (state)  
WB West Bengal (state)

Sponsors: AA6YQ

Ref. <https://groups.io/g/adifdev/message/8821>  
<https://www.iso.org/obp/ui/#iso:code:3166:IN>  
<https://groups.io/g/adifdev/message/8877>  
<https://www.iso.org/obp/ui/#iso:code:3166:CU>

### Item 148: Correct the Region enumeration's African Italy entry

Status: *Awaiting comments* *Work in progress*

In section III.B.20 Region Enumeration's African Italy, change WAE to CQ.

Ref. <https://groups.io/g/adifdev/message/8806>  
<https://groups.io/g/adifdev/message/6782>

### Item 147: Updates to Contest IDs

Status: *Work in progress*

Add the following to the Contest ID enumeration:

ARI Italian Activity Contests (IAC):

Italian Activity Contest (VHF); ARI-IAC-VHF  
Italian Activity Contest (UHF); ARI-IAC-UHF  
Italian Activity Contest (6m); ARI-IAC-6M  
Italian Activity Contest (23cm); ARI-IAC-23CM  
Italian Activity Contest (13cm+); ARI-IAC-13CM

DARC:

DARC FT4 Contest; DARC-FT4

PCC:

<https://www.procontestclub.ro/PCC%20Rules.html>  
PCC Pro CW Contest; PCC

Correct the description for CVA-DX-SSB, which says "CW" rather than "SSB".

Ref. <https://groups.io/g/adifdev/message/8804>

<https://groups.io/g/adifdev/message/8809>

<https://groups.io/g/adifdev/message/9293>

### Item 146: Provide read-only announcements of new proposed & released ADIF Specifications

Status: *Work in progress*

Consider providing read-only announcements of new proposed and released ADIF Specifications that can be subscribed to without the need to receive other traffic from the “adifdev” / “adifvoting” Groups.io groups.

Possibilities are

- Use “special” messages in the adifdev group (but requires that no other use is made of “special” messages, which cannot be guaranteed (e.g. there is more than one owner/moderator and there may be other circumstances a “special” message is needed)).
- Add a subgroup to the “adifdev” group (need to check if subgroups can be made to be announcements-only).
- Add a completely new group e.g. “adifannouncements” or “adifnotices” or ... “adifreleases” or ...

Ref. <https://groups.io/g/adifdev/message/8787>

### Item 145: Provide an optional register of PROGRAMID values

Status: *Work in progress*

Consider adding an optional register of PROGRAMID values used in the PROGRAMID and APP\_PROGRAMID\_FIELDNAME fields. This could help avoid name clashes and also be informative. This could be incorporated in the ADIF Specification, or probably better to have a website page and a link to the page in the Specification.

Ref. <https://groups.io/g/adifdev/message/8785>

### Item 144: Add download date/status fields for QRZ.com

Status: *Awaiting comments* ~~*Work in progress*~~

Add to Enumerations “QSO Download Status”

*Status, Description*

Y, the QSO has been downloaded from the online service

N, the QSO has not been downloaded from the online service

I, ignore or invalid

Add to QSO Fields:

*Field Name, Data Type, Enumeration, Description*

QRZCOM\_QSO\_DOWNLOAD\_DATE, Date, -, date QSO downloaded from QRZ.com logbook

QRZCOM\_QSO\_DOWNLOAD\_STATUS, Enumeration, QSO Download Status, QRZ.com logbook QSO download status

Sponsors: G4POP, AA6YQ

Ref. <https://groups.io/g/adifdev/message/8759>

### Item 143: Add fields for Castles on the Air

Status: Work in progress

Update 2024-07-16: This has proven problematic due to there not being a single, universally accepted list of locations and references, i.e. different countries have variations. It seems there are two approaches, neither of which is desirable: (1) have separate field for each amateur radio organization concerned or (2) have a single field that can contain distinct references for each country concerned. Although (2) seems tidier, it's not obvious that different countries' references can be confirmed to be equivalent.

t.b.s.

Ref. <https://groups.io/g/adifdev/message/7767>

### Item 135: Review the ADIF specification w.r.t. importing unexpected data

Status: Work in progress

Update 2024-07-16: No suggestions for specific changes have been made, so it's likely that the most that can be achieved is to bear this topic in mind when adding new fields or changing existing ones.

The specification rarely discusses how to deal with import of unexpected data that is conforming to a later version of ADIF. E.g. if the (MY\_)GRIDSQUARE fields had long ago specified that any characters beyond 8 should be silently ignored, then the current issue over allowing longer locators in them would not have arisen.

Ref. <https://groups.io/g/adifdev/message/8608>

### Item 129: Mode/Submode support for GMSK, FSK and FSK-W

Status: Work in progress

Add FSK as a Submode of MFSK.

Add GMSK as a Submode of PSK.

FSK-W can be logged as FSK, so I don't see need at this stage for adding an FSK-W Submode.

Ref. <https://n1mmwp.hamdocs.com/manual-operating/digital-modes/>  
<https://groups.io/g/adifdev/message/8225>

### Item 114: Register ADIF media (aka MIME) types

Status: Work in progress

2024-07-16: This has not been pursued as yet in order to avoid having mime types added then subsequently having to change them due to potential changes related to supporting international characters.

Register ADIF media (aka MIME) types.

Document them in the ADIF specification.

Ref. <https://groups.io/g/adifdev/message/8121>

### Item 112: Minor changes and corrections #5

*Status: Work in progress*

Note: the following two items were moved from Item 106

The Asiatic Russia entry:

PM Perm` (Permskaya oblast) - for contacts made on or after 2005-12-01 140 17  
30

needs changing to:

PM Perm` (Permskaya oblast) - for contacts made on or after 2005-12-01 **and made before 2011-12-02** 140 17 30

A new European Russia entry needs adding:

**PM Perm` (Permskaya oblast) - for contacts made on or after 2011-12-02 140 17 30**

The Asiatic Russia entry:

KO Republic of Komi 90 17 20

needs changing to:

KO Republic of Komi - **for contacts made before 2011-12-02** 90 17 20

A new European Russia entry needs adding:

**KO Republic of Komi – for contacts made on or after 2011-12-02 90 17 20**

“It looks like the PM, KP and KO entries should just be moved en-bloc from Asiatic Russia to European Russia without any date demarcation.”

Note: awaiting clarification from ARRL.

Ref: <https://groups.io/g/adifdev/message/7798>

## Item 107: Add Support for Extended Character Sets

*Status: Awaiting comments*

While added with the best of intentions, in retrospect the following is highly impractical for reasons including, but not limited to, the need for all the character sets to be available on every platform/runtime environment concerned, the level of skill required to program it and incompatibility with the current version of all applications that import ADIF.

[a] Add an Encoding enumeration with the following values:

Encoding, Description

WINDOWS-1252, Western European (Windows)  
US-ASCII, American Standard Code for Information Interchange  
ISO-8859-1, Latin-1 Western European  
ISO-8859-2, Latin-2 Central European  
ISO-8859-3, Latin-3 South European  
ISO-8859-4, Latin-4 North European  
ISO-8859-5, Latin/Cyrillic  
ISO-8859-6, Latin/Arabic  
ISO-8859-7, Latin/Greek

ISO-8859-8, Latin/Hebrew  
ISO-8859-9, Latin-5 Turkish  
ISO-8859-10, Latin-6 Nordic  
ISO-8859-11, Latin/Thai  
ISO-8859-13, Latin-7 Baltic Rim  
ISO-8859-14, Latin-8 Celtic  
ISO-8859-15, Latin-9  
ISO-8859-16, Latin-10 South-Eastern European

[b] Add a footnote to the above table that ISO-8859-12 does not exist.

[c] Add an ENCODING header field:

Field Name: ENCODING  
Data Type: Enumeration  
Enumeration: Encoding  
Description:

ADI files:

the character encoding used in the file.

If an application needs to convert from the encoding specified in ENCODING to a different encoding and one or more characters are unavailable in the new encoding, one option is to substitute the unavailable characters with a question mark "?".

The ENCODING field is mandatory in ADI files from version 3.2.0 onwards.

ADX files:

the ENCODING field is optional in ADX files and if present, is ignored.

[d] Change the Character type description from:

"an ASCII character whose code lies in the range of 32 through 126, inclusive"

to:

"an extended ASCII printable character whose code lies in the range of 32 through 255, inclusive.

When importing ADIF that does not contain a HEADER or whose HEADER does not specify an ENCODING field, the encoding is treated as being WINDOWS-1252."

[e] Change the description of the following fields:

PROGRAMID, USERDEFn, CALL, CONTACTED\_OP, EQ\_CALL, GUEST\_OP, OPERATOR, OWNER\_CALLSIGN, STATION\_CALLSIGN, CONTEST\_ID, SUBMODE

to add:

"Only ASCII characters whose code lies in the range of 32 through 126, inclusive, are allowed."

[f] In section "IV.A.4. Application-defined Fields" add to the definitions of PROGRAMID and FIELDNAME:

"Only ASCII characters whose code lies in the range of 32 through 126, inclusive, are allowed."

[g] In section "IV.A.6. ADI Records" change:



"If the first character of an ADI file is <, it is presumed to be the first character of the the first QSO-Data-Specifier of the first QSO Record of an ADI file that does not include a Header.

to:

"If the first character of an ADI file \*\*\*with a version earlier than 3.2.0\*\*\* is <, it is presumed to be the first character of the first QSO-Data-Specifier of the first QSO Record of an ADI file that does not include a Header."

[h] In section "IV.B.2. Application-defined Fields" add to the definition of PROGRAMID and FIELDNAME:  
"Only ASCII characters whose code lies in the range of 32 through 126, inclusive, are allowed."

[i] Add to the description of the ADIF\_VER field:

"The ADIF\_VER field is mandatory for ADIF version 3.2.0 onwards".

[j] In section "IV.A.3. ADI Header" change:

"If the first character in an ADI file is <, it contains no Header."

to:

"If the first character in an ADI file prior to version 3.2.0 is <, it contains no Header."

[k] In section "IV.A.2. ADI File Structure" change:

"...begins with an optional Header..."

to:

"...begins with a Header..."

and add:

"Note: prior to ADIF version 3.2.0, a Header is optional"

[l] Change the example in section "IV.A.3. ADI Header"

to include the new ADIF version number (3.2.0) and add an ENCODING field:

<ENCODING:12>WINDOWS-1252

[m] Change the example in section "IV.A.7. Sample ADI File" to include the new ADIF version number (3.2.0) and add an ENCODING field:

<ENCODING:12>WINDOWS-1252

Sponsors: AA6YQ, G3ZOD

Ref. <https://groups.io/g/adifdev/message/7752>

### **Item 88: Minor Changes and Corrections #3**

*□Status: Work in progress*

□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 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", a few entries are marked as "import-only" (deprecated). These should instead be marked "deleted" because they are still correct for QSOs prior to the date their use ceased.

The ADIF HTML files currently use "windows-1252" encoding; change this to UTF-8.

"We have the "Delete date" for the ON province change and "From date" for the new ON sections but there are no dates for WTX and WCF which were added in the 1980's I think (fuzzy in the mind now). Seems like there was another one that was added sometime about then but it escapes me at the moment. 73, Larry W6NWS"

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/topics/7078>

In III.C.1.b QSO Fields, for consistency with the other enumerations that have values depending on another field, change:

Submode (function of MODE field's value)

to:

(Submode, function of MODE field's value)

### Item 87: Correct HFSK submode

Status: Work in progress

In ADIF 3.0.4, HFSK was thought to be an FSK Hell mode and consequently moved from the Mode table into a submode of HELL. It's been pointed out that the FSK submode of Hell is FSKHELL. From messages some years ago, HFSK is believed to be "high speed FSK" and it is noted that then, and now, no known software specifies HFSK as a mode.

No recent progress on this due to being unable to assess the impact, i.e. how much this may have been used.

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/7018>

### Item 85: Add Primary Administrative Subdivision for Norway

Status: Awaiting comments

Add a Primary Administrative Subdivision enumeration table for the 19 fylke (counties) in Norway (see [https://en.wikipedia.org/wiki/ISO\\_3166-2:NO](https://en.wikipedia.org/wiki/ISO_3166-2:NO)).

Note that code 13 does not exist (it belonged to the city of Bergen, which was merged with 12 Hordaland in 1972).

Also the system is due to change to into 11 regioner (regions) but this is forecast to take place between 2018-2020, so the existing fylke should be added then subsequent changes can be made once the new system is in place.

Enumeration for Country Code 266 (Norway)

Code — Primary Administrative Subdivision

- 01 — Østfold
- 02 — Akershus
- 03 — Oslo
- 04 — Hedmark
- 05 — Oppland
- 06 — Buskerud
- 07 — Vestfold
- 08 — Telemark
- 09 — Aust-Agder
- 10 — Vest-Agder
- 11 — Rogaland
- 12 — Hordaland
- 14 — Sogn og Fjordane
- 15 — Møre og Romsdal
- 16 — Sør-Trøndelag
- 17 — Nord-Trøndelag
- 18 — Nordland
- 19 — Troms / Romsa (se)
- 20 — Finnmark / Finnmarku (se)

| Category      | Code | Name                 | DXCC | DXCC Name |
|---------------|------|----------------------|------|-----------|
| county        | 42   | Agder                | 266  | NORWAY    |
| county        | 34   | Innlandet            | 266  | NORWAY    |
| county        | 15   | Møre og Romsdal      | 266  | NORWAY    |
| county        | 18   | Nordland             | 266  | NORWAY    |
| county        | 03   | Oslo                 | 266  | NORWAY    |
| county        | 11   | Rogaland             | 266  | NORWAY    |
| county        | 54   | Troms og Finnmark    | 266  | NORWAY    |
| county        | 50   | Trøndelag            | 266  | NORWAY    |
| county        | 38   | Vestfold og Telemark | 266  | NORWAY    |
| county        | 46   | Vestland             | 266  | NORWAY    |
| county        | 30   | Viken                | 266  | NORWAY    |
| arctic region | 22   | Jan Mayen            | 118  | JAN MAYEN |
| arctic region | 21   | Svalbard             | 259  | SVALBARD  |

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/7084>  
<https://www.iso.org/obp/ui/#iso:code:3166:NO>  
<https://groups.io/g/adifdev/message/7054>

### Item 83: Remove Digit, Character, and IntlCharacter from Data Types table

Status: Work in progress

Background: although Digit, Character, and IntlCharacter appear in section “III.A. Data Types”, they are not used in any ADIF-defined fields nor can they be used in user-defined or application fields.

In the table in section “III.A. Data Types”:

- Remove the rows for Digit, Character, and IntlCharacter.

- Change the definition of String to:  
“a sequence of ASCII characters whose code lies in the range of 32 through 126, inclusive”
- Change the definition of MultilineString to:  
“a sequence of ASCII characters whose code lies in the range of 32 through 126, inclusive, and line-breaks, where a line break is an ASCII CR (code 13) followed immediately by an ASCII LF (code 10)”
- Change the definition of IntlString to:  
“a sequence of Unicode characters (encoded with UTF-8) excluding line break CR (code 13) and LF (code 10) characters”
- Change the definition of MultilineString to:  
“a sequence of Unicode characters (encoded with UTF-8) and line-breaks, where a line break is an ASCII CR (code 13) followed immediately by an ASCII LF (code 10)”
- In the definition of Location, change “Character” to “character” and remove the hyperlink.
- In the definition of SOTARef:  
(a) change “a sequence of Characters” to “a String”  
(b) change “a / Character” to “a / character”
- Replace occurrences of “Digit” and “Digits” by “ASCII digit” and “ASCII digits” and remove the hyperlinks.

In section “IV.A.1. ADI Data-Specifiers”, remove “IntlCharacter,” from “Fields of type IntlCharacter, IntlString, and IntlMultilineString cannot be used in ADI files.”

In section “IV.A.4. Application-defined Fields”, remove:  
“, which can be for any Data type except Digit and Character”

In section “IV.A.5. User-defined Fields”, remove:  
“, which can be for any Data type except Digit and Character”

In section “IV.B.2. Application-defined Fields”, remove:  
“, which can be for any Data type except Digit, Character, or IntlCharacter”

In section “IV.B.3. User-defined Fields”, remove:  
“, which can be for any Data type except Digit, Character, or IntlCharacter”

Ref. tbs

## **Item 82: Add section describing the meaning of Deleted**

Status: Included in 3.1.4

Add a new section that will appear after the “Deprecation” section:

### **"II.C. Deletion**

Deleted enumeration values are those that are only valid for QSOs that occurred within a specified time range in the past. They shall only be accepted when importing QSOs that occurred within that time range, and shall only be emitted when exporting QSOs that occurred within that time range."

Ref. tbs

## **Item 81: Replace “Y” in Deleted columns with “Deleted”**

Status: Awaiting comments

A number of tables in the ADIF specification have a "Deleted" column that are blank or have the value "Y" in for a deleted item.

Change the "Y" values to "Deleted" so that they are more prominent, especially for larger tables that flow over more than one page / screen.

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6837>

## Item 80: Clarifications on the fields BAND, BAND\_RX, FREQ, & FREQ\_RX

*□Status: Work in progress*

(1) Add the examples <FREQ:5>7.035 to the FREQ and FREQ\_RX field descriptions and <BAND:3>40m to the BAND and BAND\_RX field descriptions.

(2) Add the following at the start of the Band enumeration section:

"Although the FREQ and BAND fields are discussed in this section, the same considerations apply to the FREQ\_RX and BAND\_RX fields.

FREQ contains a frequency in MHz and must only contain digits and at most one decimal point character '.' (ASCII code 46). In particular, NOT ALLOWED is the use of a comma character instead of a '.' and NOT ALLOWED is the use of thousands separators.

BAND contains a band from the Band enumeration below. It must not contain spaces.

The following are guidelines for use of band and frequency in records representing QSOs.

(a) If both FREQ and BAND fields are exported, their values must be consistent according to the values in the Band enumeration table in this section.

(b) If in an imported record:

- FREQ and BAND are inconsistent, use FREQ and ignore BAND.
- FREQ is invalid and BAND is present, use BAND and ignore FREQ.
- FREQ is invalid and BAND is absent or invalid, reject or ignore the record.
- BAND is invalid and FREQ is present, use FREQ and ignore BAND.
- BAND is invalid and FREQ is absent or invalid, reject or ignore the record.

(c) Applications should try to help users avoid mistakenly recording VHF and higher frequencies when they intended HF; this can happen when frequencies in different bands are multiples of 10 apart and can be entered erroneously due to the difference being in the position of a decimal point. For example, 10.105000 MHz (30m) and 10105.000 MHz (3cm)."

Ref. <https://groups.yahoo.com/neo/groups/adifdev/conversations/messages/6995>

## Item 71: Make ADIF test files available

*Status: Work in progress*

Update 2024-07-16: Test files, although not perfect, are provided for each new release of ADIF and documented in the Groups.io group. The reason that their existence is not yet incorporated in the ADIF

specification / resource documents is the desirability of tidying up the software prior to making it available on GitHub.

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

A 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 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.) These files can also include one field per record to further test the structure.

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 adif.org.uk website and the ADIF Development (ADIFDev) Yahoo group’s Files area.

Limitation: applications being tested should 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 70: Make files exported from the ADIF Specification available

*Status: Work in progress*

Update 2024-07-16: Exported files are provided for each new release of ADIF and documented in the Groups.io group. The reason that their existence is not yet incorporated in the ADIF specification / resource documents is the desirability of tidying up the software prior to making it available on GitHub.

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.

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 date data type <https://www.w3.org/TR/xmlschema-2/#date> UTC values, e.g.  
2017-05-22Z

#### **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".

In the files containing fields, where a field's value is taken from a set of enumerations depending on the value of a different field, the enumeration name will be suffixed by '[' followed by the name of the other field followed by ']'. E.g. for the STATE field, the specific Primary\_Administrative\_Subdivision enumeration depends on the value of the DXCC field, so the enumeration name will be included as:

Primary\_Administrative\_Subdivision[DXCC]

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.

The Sponsored Award enumeration is not exported because its values are not defined within the ADIF Specification. However, the table of sponsor names is exported with the name "Award\_Sponsor".

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.

Similarly, any Secondary 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 Initialize method is run, three message boxes will be displayed 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 greater than or minimum, and / or maximum values specified in their descriptions have the values surrounded by <span> tags with the title attribute set to "GreaterThan", "Minimum", or "Maximum" e.g. <span title="Maximum">99999999</span> GreaterThan is used where the text in the Specification feels more natural using it rather than a Minimum value. The GreaterThan value must be an integer and is converted to a Minimum value by adding 1. "GreaterThan" is not currently used with values containing a decimal point; "Minimum" can be used instead.

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 65: Add a list of US Counties

*Status: Work in progress*

Update 2024-07-16: No consensus was reached on this proposal for several reasons (although arguably the first item below is the main obstacle):

- At the time the proposal was raised, the Counties Award sponsor did not provide an electronic list of counties (and an attempt to obtain a paper one by making a payment was declined due to long-term unavailability).
- The publicly-available FIPS 6-4 document ceased to be provided in 2002.
- When this topic was last looked at, the replacement for FIPS 6-4 was an ANSI document that had to be paid for, making it an undesirable resource for amateur radio.
- There is a Wikipedia list but Wikipedia is usually not considered as an authoritative reference for ADIF: [https://en.wikipedia.org/wiki/List\\_of\\_United\\_States\\_counties\\_and\\_county\\_equivalents](https://en.wikipedia.org/wiki/List_of_United_States_counties_and_county_equivalents)
- Adding a list to ADIF would result in problems mapping whatever county names are currently in use in ADIF into the proposed ADIF list.

t.b.s.

Ref.

## Item 62.2: Update Primary Administrative Subdivision for Country Code 224 (Finland)

*Status: Ready for inclusion*

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

[https://groups.io/g/adifdev/files/Enumeration\\_for\\_country\\_code\\_224\\_%28Finland%29\\_2017-03-16.csv](https://groups.io/g/adifdev/files/Enumeration_for_country_code_224_%28Finland%29_2017-03-16.csv)

[https://groups.io/g/adifdev/files/Enumeration\\_for\\_country\\_code\\_5\\_%28Aland\\_Is.%29\\_2017-03-16.csv](https://groups.io/g/adifdev/files/Enumeration_for_country_code_5_%28Aland_Is.%29_2017-03-16.csv)

[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_for_country_code_224_(Finland)_2017-03-16.csv)

Ref. <https://groups.io/g/adifdev/message/6764>

<https://www.sral.fi/sv/hobbyn/infobanken/kommunforteckning/>

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

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

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

*Status: Work in progress*

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 47: Add JARL awards to Credit enumeration

*Status: Work in progress*

Update 2024-07-16: There was a failure to agree that the following was a workable implementation. As the topic has not been raised for a long time, this item will be removed in the next version of this document.

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 11: Add REP DMP Award

*Status: Awaiting sponsors*

Update 2024-07-16: No sponsors have come forward so this will be removed at the next version of this document.

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>