The Pre-History of Contest Software

This article is a wee bit retro, but contains important suggestions that would enable a smoother data entry of callsign and exchange information (especially corrections) in all contests. I am Roy, AD5Q, and I have been programming computers since 1967. Through the 1970's this was primarily on mainframes writing accounting systems in assembly language, and in 1980 I switched to programming PC's. As the ARRL DX CW Contest approached in February 1981, I had an idea for a program that would log, dupe and score the contest in real time. I wrote the prototype in less than a week and used it in the contest. It was a dramatic departure from paper logs and dupe sheets. I accumulated fairly decent QSO totals with ease. It was a beginning. This was, beyond any doubt, the way to go.

I don't remember if I sent in the logs for that first contest. Development continued, mostly during 1981-82. There was debugging to do during the contests (sometimes after many hours of no sleep), and sometimes I lost data (DX contacts & mults). When the data was corrupted, I didn't send the logs in. I know that the first version of the program, the prototype, did not compute the run rate. This was added for the IARU in July (back then it was a different contest called the "Radiosport"). There were a few differences in the statistics I displayed, such as overall rates per band, and total time spent on each band. My short term QSO rate was computed based on the most recent 15 minutes, instead of the last 10 QSOs. As with all software, I kept adding features. My original intention was to offer the software package to contesters at a fair price and make a modest living at it.

My point here, is that I don't think anyone else was doing this back then. Relevant software articles in NCJ were mostly about programs you would use AFTER a contest, such as to enter the callsigns worked on each band and produce the required hard copy dupe sheets. I could be wrong, but I think I was the first to have a program that logged contest QSO's at competitive rates, duped, decoded the callsign to determine the country mult, continent and QSO points, displayed mult totals for each band, and therefore scored the contest - all in real time. If you know of someone who did this prior to 1981, please write.

At that time, radio hams and contesters did have computers. Most commonly they were Commodore 64's. I decided that this platform was unsuitable for contesting for two reasons: 1) the keyboard had a spongy feel and was not good for accurate typing. 2) there was no system clock for logging the QSO time. The original version of my program ran on a Radio Shack TRS-80 Model II (their serious small business PC at the time) using their proprietary operating system (TRSDOS) and interpretive BASIC language. This platform had limitations. As you may already know, Radio Shack sucks. W2NSD published a magazine dedicated to their computers, and Radio Shack refused to carry it in their stores because he was critical of the manufacturer and its policies.

After logging several hundred QSO's the limitations became obvious. String space became constricted, slowing processing and ulitmately forcing a long pause for "garbage collection". With over 1,000 contacts in the log, this would take up to 2 minutes AND become more and more frequent: unacceptable. By CQWW SSB in 1981 I had switched the operating system to Pickles & Trout CP/M, and re-written the program in a compiled version of Digital Research CBASIC called CB80. Also, the callsign dupe table was handled in Z-80 assembly language within a block of memory instead of with BASIC strings. This eliminated the Garbage Collection pauses, greatly increased the speed, and enabled me to develop for any Z-80 PC running CP/M. This was at the time the standard platform for small business computing. By mid-1982 I had separate modules for the ARRL DX Contest (either DXCC or stateside mults), WPX, RadioSport (aka IARU), CQWW, Field Day, CQ/M and the 10M Contest. I also had a more general database for everyday QSO's where I could combine them with all the contest logs and do DXCC tracking and QSL's. I eventually added modules for the NA Sprint and 160M contests. All these old logs are in my current DX4Win database.

I continued to polish up the program, tweak out the remaining bugs, and was getting ready for the great migration of contesters from their Commodore 64's to the CP/M platform (which didn't happen). There were challenges. The maximum memory on a Z-80 PC was 64k. There were no standard hardware calls in CP/M PC's for things such as cursor positioning on monitors, no standard keyboards, and no standard hex values returned for keystrokes such as arrow keys. All software in those days had configuration routines for the monitor and keyboard. The main challenge for me, however, was the CPU clock. I had to access it using PEEKs and POKEs into upper memory, and first had to find the memory locations where something was ticking away the minutes and seconds. It was an overnight run. Once it found the clock in memory, an algorithm had to determine the format: decimal or hex seconds, minutes, & hours – or a single hex value for the time, in seconds. Only then would I have a configuration setting that controlled how I would read and write to the system clock on a particular PC with PEEK & POKE. These early clocks also drifted a lot, so there was another long run where I would calibrate the clock against WWV, and apply an offset correcting for the drift every time a QSO was saved.

All for nothing. Doctor DX was released, and it somehow ran on a Commodore and did the CQWW! The contesters on the local repeater loved it, but I was devastated because it sounded like I had competition. I then figured out that it was just a toy simulator and didn't actually log the contest. This was in late 1983 I think. But somewhere during 1983 and '84 I came to the conclusion that there was so much software copying going on that I was not going to make back my development costs or any profit by distributing the software. At the time, all software was distributed on floppy, and cracking the copy protection schemes on floppy disks was great sport. I would sell maybe 4 or 5 copies, and everyone would have my program. I found something else to do, development stopped, and you never heard of my programs. To this day, I don't think anyone has made back their development costs on logging software of any kind.

And so, for at least the first half of the 80's I did all my contest logging using software, while everyone else continued to log and dupe with pencil and paper unless they also wrote their own code. There was no software for sale that a serious contester would want. In the rare on-air SSB QSO's I had with other contesters, most didn't even believe me when I told them what my software did. Eventually, CT was released. It was for MSDOS, not CP/M, and it was a better program anyway (but buggier). NA followed, and I was looking forward to migrating to them myself. They did some neat things that my software didn't. They interfaced to the radio, and they sent CW. Hey, the screens were even in color!

Huh?

My software never sent CW.

If you have read this far, take a break. Go ahead and roll on the floor laughing. But keep reading. We are getting to the fun part.

No F1 key? AutoCQ? No INS/PLUS. No callsign key. No ESM (Enter Sends Message). None of that. I am not saying things were better that way, but we are now leading to the main reason I am writing THIS article at THIS time: SMOOTHER DATA ENTRY. The single feature of my archaic software that would be a widely appreciated improvement to modern contest software such as WriteLog, N1MM and Win-Test; is the way I did the data entry for the exchange – especially callsigns and fills. The rest of my program truly belongs in the trash.

Throughout the 80's and into the early 90's, I did ALL my sending by hand with a paddle and only a WB4VVF style Accu-Keyer. There was NO memory, even for a CQ. Thus, I operated entire contests, sending every single CQ, and every single dit and dah – BY HAND, while logging with computer. This means, that every minute of any run – even really slow runs – I was fully occupied either sending or entering data. There was no such thing as making corrections to the log while a CQ message was being sent automatically.

My station during this period was a tribander & wires. First a tri-band HF Quagi at 70 ft, and after 1987 a KT34XA at 104 ft. I converted the software to MSDOS around 1987, still using Digital Research BASIC (CB86). I had some fairly respectable scores, but had a goal of 2000+ QSOs by hand keyer. CT was finally released later in the '80s, it was a better program than mine, but I really wanted to top 2000 Q's by hand before making the switch. There were several contests, including 2 ARRL CW's, where I had over 1,800 Q's by paddle only. So I guess that was my limit. In any case, the data entry scheme I used for thousands of QSOs was ultra smooth and proven by time – making corrections to callsigns and exchange information very easy. It had to be, because the demands of manual sending left no slack whatsoever.

My first hands-on experience with CT wasn't until my first multi-op invitation: the 1992 CQWW at AA6TT. I made one more attempt at 2000 Q's with my own stuff after that, the 1993 WPX, and finally switched to NA. When I saw that CT and NA were tabbing from field to field to do data entry, I was R.O.T.F.L. I instantly saw the special problem they would encounter regarding the ARRL Sweepstakes. And I had the solution all along. I was not about to upgrade my old monochrome software and compete with these products (especially since I was no longer interested in programming in any form of BASIC), but felt it was a good idea to make a contribution to the state of the art. So I wrote an article for the NCJ detailing the way I had been doing data entry since 1981. The emphasis of the article was on the SS exchange, and also on callsign entry and any other exchange info. It was an approach that encompassed ALL data exchange and callsign entry for ALL contests.

The article was rejected by the NCJ editor as “too arcane” for publication. Hmmmph. I didn't quit there, and ultimately mailed hard copies (this was before Internet) to 4 other people – all very well-known calls. Two of them were the authors of NA and CT (TR didn't exist yet). I don't know what happened after that, or if anyone communicated with other software authors. It seemed that everyone just blew me off. I am very used to that. I truly am too arcane for NCJ. It's not something I can change. I am autistic, but with an IQ of 162. It's called Asperger Syndrome, and it's the reason I rarely

work SSB or 2 Meters, or show up at social functions like Dayton and Visalia. I am mostly out of the loop, and deeply involved in a wide spectrum of interests that are too arcane to discuss on the ham bands. Excuse me here, but I needed to say that. So you know. I have issues :-D

I felt vindicated when a new program, TR, incorporated my suggestions regarding the SS exchange. Though I suspected that somebody forwarded my article to its author, I don't really know and have never asked. It's extremely likely that he arrived at the solution independently. The problem with tabbing all the fields in the SS exchange is obvious, and so is the parsing solution. It became the most widely promoted feature in his program for several years running, and it's very popular. I bought a copy of TRLog and tried hard to use it (and also all the others at some point), but I've never used it in the SS. There are other things about TR that I very much don't like, and which will not be referenced in this post. I liked NA much better for SO2R, which is probably true of anyone who has tried it.

There were other suggestions in my 1980's article that were not implemented in TR, and I think they would also be popular now regardless of whose software adopted them. They are based on the same principle as the so-called "TRLog Method" of breaking down the SS exchange based on the syntax of the data, and were originally deployed in my own software for the 1981 fall season and all supported contests thereafter. It is the continued popularity of the TR formatted SS exchange in Windows software that encourages me to write this documentation, as I am suggesting that the technique be expanded to support all contests, and include a more flexible entry of callsigns.

Unlike TR, my method used a single field for the entry of all contest data. TR has a field for the callsign in the SS, and another field for everything else (including corrections to the callsign). The callsign field is not necessary. Now you no longer worry about what field you are in at all. There is a single prompt character, followed by an area to enter anything in any order. Above the entry bucket, there is a 2nd line that is formatted as it will appear in the log. This is for the current QSO. Recently entered QSOs can be shown above the current QSO line, the log can be scrolled, and saved QSOs can be edited with additional corrections. Nowadays this is standard stuff. (My software didn't have this though. Once the QSO was saved, it was gone from the screen. No edits after the QSO, only during. I am not suggesting we go back to that.)

The underlying principle for all this data entry is that it is much easier to type a correction than it is to backspace, press the home key, delete bad characters in busted information, and THEN type the correction. Does that make sense? For the most part, forget about backspacing, tabbing and deleting. Just type fills until it's right. Corrections will automatically overwrite any busted information without the need to tab or position the cursor. Examples follow.

For the purpose of these illustrations, let's use a colon as the prompt character. Also, let's call the individual elements of the exchange "chunks", which are separated from other "chunks" by spaces. The logic will determine if a chunk contains a full callsign, or just a prefix or suffix, or maybe it's part of the exchange. We enter what we copy, and the software figures out what to do with it.

Most of the examples below will begin with a busted call, then show the chunks needed to correct it. A prefix is everything up to the last numeric, and a suffix is everything beyond it. Thus, when correcting a suffix, everything following the last numeric is replaced with the new suffix chunk. To correct a prefix, everything up to the last numeric is replaced. The chunks with slashes are appended before or after the main call, depending on the position of the slash within the chunk.

: W8GT GM N8 VP9/

. A callsign consists of a prefix, a suffix and MAYBE something with a slash before, after, or both.
. A full callsign is a chunk containing both letters and numbers, and ending with a letter.
. A suffix is a chunk containing only letters.
. A prefix is a chunk containing both letters and numbers, and ending with a number.
. If a chunk ends in a slash, the chunk will preceed the full call after formatting.
. If a chunk begins in a slash, it will follow the full call.
. If a chunk consists of just a single slash (rare), we want to delete the portable section of a call
. When you press ENTER, the contents of the whole entry bucket is processed and the fields are formatted into the log line above it. The prompt line is cleared, but ready for additional input to the current QSO before saving.
. When you press ENTER on the empty prompt line, the QSO is saved and you send your TU QRZ message and update score, totals, rates, etc.

1 0001z VP9/N8GM 599 599 05
: K8 VP8/

We have just processed the first example, concluding that the call is VP9/N8GM. We have also guessed the exchange for CQWW as zone 05 if you want to do that. Now the prompt line contains two additional corrections. We press ENTER to correct the prefix to VP8/K8GM, and the guessed exchange should change to zone 13.

```
  1 0001z VP8/K8GM        599 599 13
: /
```

Now, suppose we want to change the above call to simply K8GM, deleting the VP8 part (rarely needed, but supported). Enter a single slash.

```
  1 0001z K8GM            599 599 13
:
```

We can keep entering corrections and pressing ENTER to format them into their position in the callsign. When we are done, we press ENTER without any corrections. This saves the QSO (and if you are programming with ESM, this would also send the TU QRZ type message). This entry method WILL effect the way you program ESM, but it should work well.

In a contest like the CQWW, the guessed zone might be wrong with some stations, especially in the USA. Just type the 2 digit zone, or maybe allow a single digit for the zone.

The CQWW is the simplest case, because most of the time you enter only callsigns. Let's try this in the IARU. This is also a zoned contest, but with a difference;

```
: HG1S HQ MRASZ
```

This will initially correct the callsign HG1S to HG1HQ, but will next correct it again to HG1MRASZ. What's wrong? We need a way to tell the program that MRASZ is not a suffix. We use a punctuation mark to distinguish between a callsign suffix and the mult field. My program used to use a period, but it should be user selectable (contest configuration). I suggest something very near the ENTER key, and for the purpose of this and further illustration, let's use the apostrophe:

```
: HG1S HQ MRASZ'
```

This should process as a completed exchange:

```
  1 0001z VP8/K8GM        599 599 13
  2 0003z HG1HQ           599 599 MRASZ
```

In the above, I “tagged” the MRASZ mult field with an apostrophe after the mult string. The tag character is a minor inconvenience that allows us to extend the flexible callsign entry methodology outlined above to practically any contest. It is used primarily to mark non-numeric chunks that aren't callsign suffixes. In the IARU, they are used only in anomalous situations (HQ stations). In others, they are part of every exchange. The benefits of entering QSO and callsign information in any order without cursor positioning make this an easy tradeoff. This tag character can be entered either immediately BEFORE or AFTER the exchange data. I always accepted and recognized it either way. In other words, you should NOT attempt to program an exchange where a chunk beginning with the tag character implies a different part of the exchange than one with the tag character at the end. In practice, you will get them mixed up.

Here is another anomalous situation, the ONLY time you will need the tag character in the ARRL DX Contest if you are US/VE:

```
: OH2BH 'KW
```

But if you are DX in this contest, the exchange consists of states and provinces and you need to mark them as non-suffixes:

```
: K3LR 'PA
```

If the exchange is a serial number or any other numeric, just enter it. Anything that is all numeric is NEVER part of the callsign.

In the Ten Meter Contest, the exchange can be a number (DX) or a US/VE state/province, which consists of characters. The numeric is no problem. The state/province needs the tag character, either before or after:

: K5ZB 'MA ZD
It's Randy.

I have entered a few thousand contacts in the 10M Contest this way. I never had a problem remembering to include the tag character, either on CW or SSB. If the station is US or VE, you know the exchange will need it. When working a stateside station in this contest, I would typically type the tag character while copying the signal report, knowing that the state is sent next. Easy.

This data entry is extremely smooth on CW or SSB, regardless of rate. On Phone, many stations use the lasstoo (last two) letters of their call. Just enter what they say. When they follow up with their full calls, you can enter just the prefix if the suffix really was only 2 letters. Copying the suffix first and the prefix last is no problem at all. No backspacing or positioning. The logic will sort it out. Done!

Here are a couple more examples:
: EW5GN EI5/ EI/
It's Barry.
: HB9BRV HB0/ DRV /P
It's HB0/HB9DRV/P.

Of course, you can ALWAYS enter full calls.

Now, let's look at the NAQP. It's a problem.

: K5NA GA BILLY TX

Here, the suffix, name and state are all letters and the logic will treat all as suffixes unless we do three things:
. Require full calls in this contest.
. Identify either the name or state (I would use the state) with the tag character.
. Any chunk not containing a numeric or the tag character is the Name.

Therefore:

: K5NA K5GA BILLY 'TX
Does it right.

The NA Sprint is similar, but there is also a QSO number in the exchange. So the above QSO might look like this:

:K5NA K5GA 315 BILLY 'TX

So here is the rule regarding the use of the "tag" character and support for flexible callsign entry:

. In contests where all parts of the exchange are numeric, you may use my more flexible approach to entering callsigns. This means you normally need to type only the portion of the callsign that needs a fill.
. If the contest exchange includes a component that contains no numerics, you need the tag character to distinguish it from a suffix.
. If the exchange includes TWO components without numerics, then ONE of them needs to be identified with the tag character. Flexible callsign entry (of prefixes, suffixes, etc) needs to be disabled, otherwise the other part of the exchange will be treated as a call suffix. You need to always enter the full callsign :-(

We already know about the SS Exchange from TRLog. Here, 2 digit numerics are checks and others are numbers. Following a 2 digit numeric with the single character precedent has the effect of distinguishing the number from the 2 digit check. Prefixes should work fine because none of the other parts of the exchange end in a numeric. I've noticed that TR allows the entry of suffixes, but that's tricky because some call suffixes are valid section ID's. Be careful with that. Most of the time we just enter 123B and 64STX and it works fine. An alternative that would clear up the conflict between suffix and section would involve the tag character, and would only be needed when we depart from the 123B and 64STX

format to enter a component by itself. Use it for 2 digit numbers, single character precedents AND multiple character sections. The benefit is that you would be able to enter prefixes and suffixes of callsigns with impunity. Full support of partial callsigns with "TR style" SS exchange.

Any correction to the callsign should also correct the CW message sent when the worked station's callsign key is pressed (TRLog didn't do that).

There are rare instances where stations will send you an honest signal report (other than 5NN) in a contest. Since the signal report is part of the exchange, it is important to log what is sent. Since this is a rare event, we can use a more inaccessable character of the keyboard to tag the report. I suggest something requiring the shift key, such as the colon. Thus:

: 579:

Or, if you want to get fancy and respond with an honest report...

: 579R: 559S:

… where the R and S indicate received and sent reports. If you choose to support the entry of a SENT signal report other than 5NN, however, note that this would effect your CW messages. You would need a new variable for users to insert everywhere in their messages in place of "5NN". The variable would certainly default to 5NN. In Win-Test, for example, it would be something like "$SENTRPT". I think this is optional. Since signal reports other than 5NN are so rare, you may not want to support this in your SENT report, but received reports need to be copied as sent and logged.

You might think this is funny, but I also had a chunk in the following format: "1437Z". This would re-sync my CPU clock to WWV in mid-contest. The feature was necessary back then, because the clocks did drift. In the CP/M standard, a CPU clock wasn't even required and some computers didn't have them.

In all the above examples, Tabs, Home, End, Arrow Keys, Delete and BackSpace keys are usually not needed, but they should definitely function normally. Once there are no corrections, just press ENTER to save the QSO. This means that if you know you have entered an exchange correctly and wish to finish the QSO, then you will need to hit ENTER twice. Pressing it just once allows for the entry of additional fills. You might additionally program the PLUS key to finish the QSO, and press it just once. Each contest is a little different, but with a few configuration parameters it should be possible to quickly set up any contest to parse the exchange and callsign info correctly.

Contesting software has moved beyond DOS (and CP/M), but the cult following behind the popularity of TRLog still seems to have much influence over the new programs. Some are using the fine LUA scripting capability of Win-Test to replicate the functionality of TR, even to the extreme of using the same keystrokes. This is retrograde. Both N1MM and Win-Test have duplicated the TR style of entering the SS, but neither have expanded the underlying concept (of using the syntax of the data to identify parts of the exchange) to callsign entry and across all other contest exchanges. In this regard, software has still not caught up to where I was in 1981 with just a floppy driven Trash-80 Model II & CP/M. To this end, I have documented my algorithm and suggest we all play with it some. Good luck. I already know that it works really well.

I will leave it to the programmers to get the ESM messaging working correctly on CW. I really hope somebody does this, especially the Win-Test team. It shouldn't be hard to code, but these entry capabilities need to be embedded into the main program, replacing completely the tabbed entry with a single field for entering data in any order – NOT entered in a separate box with LUA.

Roy – AD5Q
Houston Tx
9 Nov 2013