VKV **SKIMMER**

USER GUIDE · VERSION 1.5.19

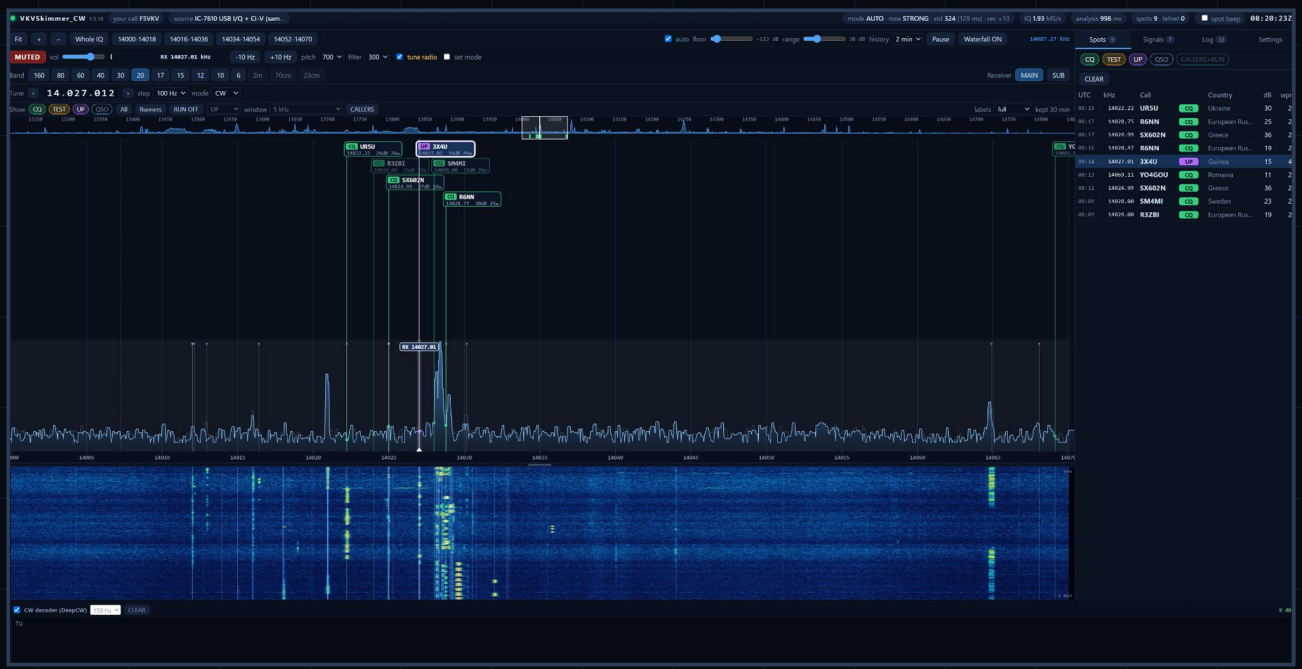
VKVSkimmer

CW and RTTY skimmer

Installation · Settings · The GUI · RUN mode

Telnet cluster · N1MM+ spectrum · RBN Aggregator · RBN Reference

Full list of settings (skimmer.ini / rtty.ini)



https://vkvskimmer.com · Free public release · Fabio Roccatagliata, F5VKV

Contents

1. What you get 3
2. Installation 4
3. First start: callsign and input source 5
4. The GUI 7
5. RUN mode 9
6. Telnet cluster 11
7. Spectrum in N1MM+ 12
8. Sending spots to the RBN (RBN Aggregator) 14
9. RBN Reference (incoming RBN spots) 18
10. Contests 19
11. Troubleshooting 20
Appendix A. All settings of skimmer.ini (CW) 21
Appendix B. rtty.ini: what is different 27

1. What you get

VKVSkimmer is distributed as two independent Windows programs: **VKVSkimmer_CW.exe** and **VKVSkimmer_RTTY.exe**. Each has its own decoder chain, its own INI file and its own ports, and you can run one or both. The browser GUI, the telnet cluster and the integrations are the same, so everything in this guide applies to both.

	CW skimmer	RTTY skimmer
Program	VKVSkimmer_CW.exe	VKVSkimmer_RTTY.exe
Start with logs (optional)	VKVSkimmer_CW_with_logs.bat	VKVSkimmer_RTTY_with_logs.bat
Settings file	skimmer.ini	rtty.ini
Browser GUI	http://127.0.0.1:8073/	http://127.0.0.1:8074/
Telnet cluster (spots)	port 7300	port 7301
Telnet HEARD stream	port 7310	port 7311
RBN Aggregator port	off (rbn_port = 0)	off (rbn_port = 0)

The GUI opens by itself in your default browser when the program starts. Public releases are free: no licence file, no expiry.



The browser GUI: band and view controls at the top, spectrum with spot labels, waterfall, live decoder at the bottom, Spots / Signals / Log / Settings on the right.

2. Installation

1. Unzip the whole package into one folder, for example `C:\VKVSkimmer`. Keep every file and the `models` folder next to the programs.
2. Install the software for your input source (below). Only for the input you use.
3. Open `skimmer.ini` (CW) and/or `rtty.ini` (RTTY) and write your callsign: `callsign = F5VKV`. You can also do it later in the GUI.
4. Start `VKVSkimmer_CW.exe` or `VKVSkimmer_RTTY.exe`. The GUI opens in the browser. To produce logs (for example to report a problem), start `VKVSkimmer_CW_with_logs.bat` or `VKVSkimmer_RTTY_with_logs.bat` instead.

Requirements: Windows 64-bit and a modern browser (Edge, Chrome or Firefox). An internet connection is recommended for the daily MASTER.SCP / cty.dat updates and for RBN Reference; the skimmer works without it.

Icom IC-7610 / IC-7760 - direct I/Q

VKVSkimmer reads the I/Q directly from the radio's USB 3 I/Q port, in 64-bit: no 32-bit Icom ExtIO driver, no HDSDR, no virtual audio cables.

- IC-7610: firmware 1.20 or later (needed for the I/Q output).
- From the Icom download page of your radio install the **USB I/Q Package for HDSDR**: it installs the driver of the [USB 2] I/Q port. HDSDR itself is not needed.
- Connect the radio's [USB 2] port to a USB 3.0 port of the PC with a real USB 3.0 cable.
- `FTD3XX.dll` is included: keep it next to the programs.
- CI-V runs over the same USB I/Q interface: the radio's other USB / COM CI-V port stays free for your logger.

SDRplay RSP

- Install the SDRplay **Hardware API 3.x** for Windows from sdrplay.com. It installs `sdrplay_api.dll` and the SDRplay API service, which must be running.
- Supported: RSP1, RSP1A, RSP1B, RSP2, RSPdx, RSPduo (single tuner).
- Close SDRuno, SDRconnect or any other program that uses the RSP before starting the skimmer.

RTL-SDR dongle

- Download the RTL-SDR Blog Windows release (`Release.zip`) from github.com/rtlsdrblog/rtl-sdr-blog/releases and copy `rtl_tcp.exe` and the DLL files of its `x64` folder next to the programs. The skimmer starts and stops rtl_tcp by itself.
- Install the WinUSB driver with Zadig (zadig.akeo.ie): run it as administrator, Options > List All Devices, select **Bulk-In, Interface (Interface 0)**, WinUSB, Replace Driver.
- On HF with an RTL-SDR Blog V3 the skimmer uses direct sampling (Q branch) by itself (`rtl_direct = auto`).
- Close SDR#, SDR++ or any other program that uses the dongle before starting the skimmer.

Sound card

No extra software: receiver audio of any radio, I/Q from an SDR on a sound card input, or the IF output of an Icom radio (USB audio codec, about 15 kHz).

3. First start: callsign and input source

Open the **Settings** tab on the right of the GUI. Each block has a badge: **live** = applied at once; **restart** = saved to the INI file, applied at the next start.



Settings: input source, spectrum output for the logger, CPU profiles, diagnostics and your callsign.

Your callsign

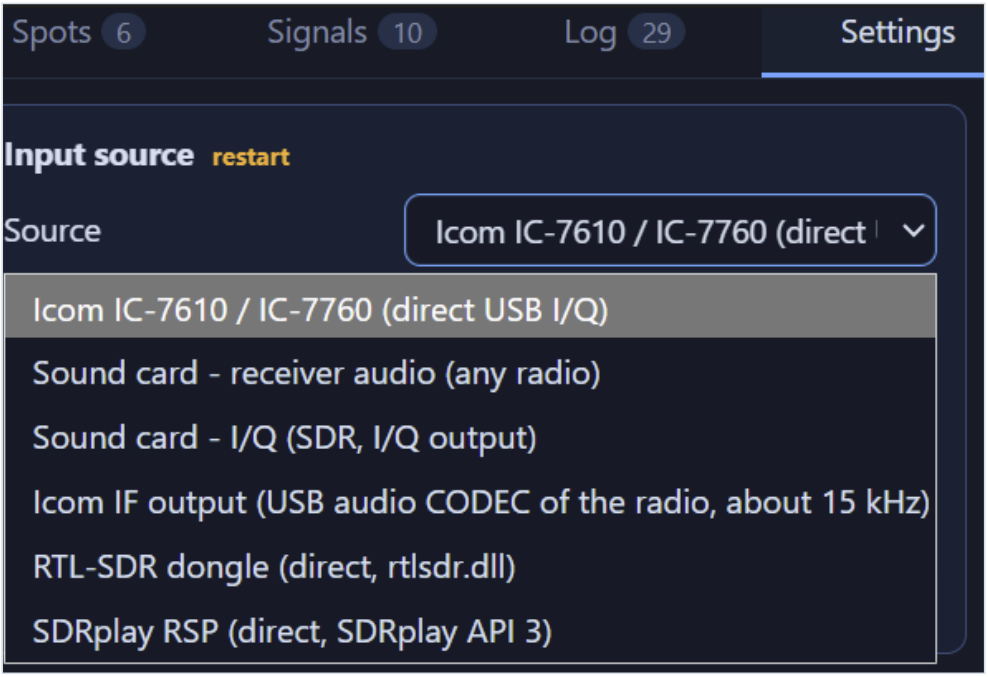
Settings > Your station > **Your callsign** (live). It is the spotter of every line (`DX de F5VKV-#`), the telnet greeting, the RBN Aggregator banner and the RBN login.

> **REQUIRED**
> Without a valid callsign the skimmer decodes and shows the stations in the GUI, but publishes **nothing**: no telnet, no HEARD stream, no RBN Aggregator. A red **SET YOUR CALLSIGN** warning appears in the top bar.

Portable calls are accepted with a real operating prefix (`F/F5VKV`, checked against cty.dat) or a suffix (`F5VKV/P`).

Input source

Settings > **Input source**, choose the source, **Save input**, then restart the program. The line *in use now* shows what the program is actually using.



The input sources.

Source	INI value	Notes
Icom IC-7610 / IC-7760 (direct USB I/Q)	`input = icom`	Whole I/Q band, CI-V tuning, MAIN/SUB, RUN mode, TX protection
Sound card - receiver audio (any radio)	`input = sound_audio`	Audio of the receiver; set sound_rf_khz to the dial frequency
Sound card - I/Q (SDR, I/Q output)	`input = sound_iq`	I/Q of an SDR on a stereo input
Icom IF output (USB audio codec)	`input = sound_if`	About 15 kHz around the dial
RTL-SDR dongle	`input = rtlsdr`	Through rtl_tcp (RTL-SDR Blog release)
SDRplay RSP	`input = sdrplay`	SDRplay API 3

CPU profile (CW)

Settings > **Profiles** (restart): **AUTO** (recommended: adapts the decoding effort to the CPU load), **LIGHT** (older or low-power PCs), **MAXIMUM** (powerful PCs). The **Diagnostics** block shows the signal rate, lost data and the analysis time per step: if *Lost data* is not *none*, choose LIGHT.

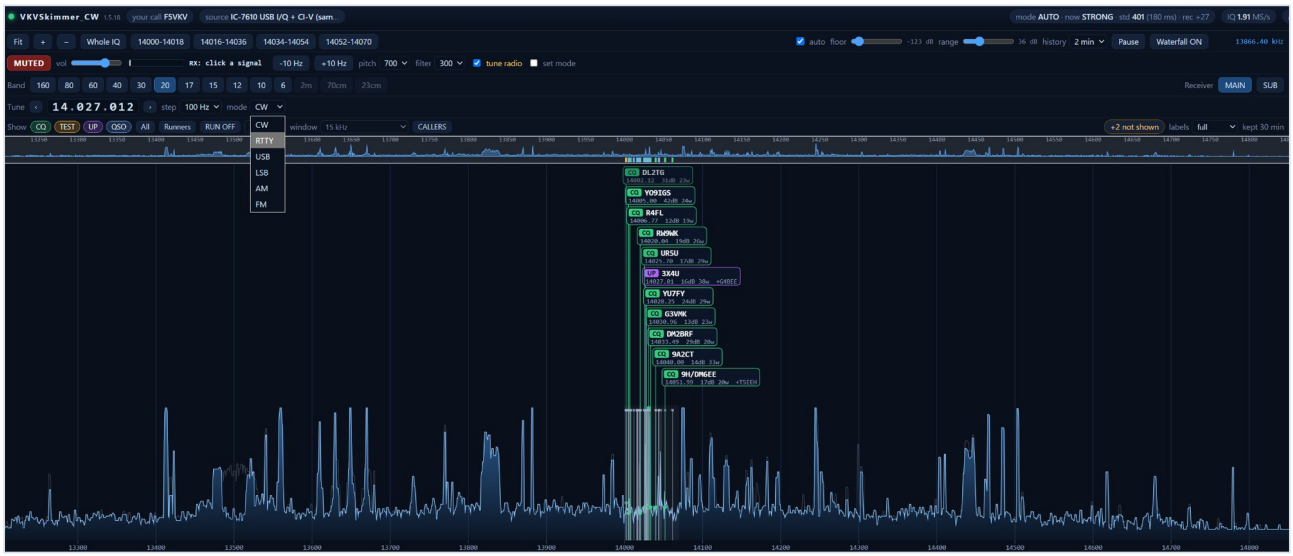
4. The GUI

Area	What it does
Top bar	Your call, input source, decoder mode, I/Q rate, analysis time, number of spots and telnet clients, spot beep, UTC clock
Fit / + / - / Whole IQ	Zoom on the analysed segments, zoom in and out, or show the whole I/Q band of the source
Segment buttons	Jump to one analysed segment (e.g. 14000-14018)
Band / Receiver	Band buttons (the radio follows), MAIN / SUB receiver of the Icom
Tune / mode	Frequency of the receiver you listen to and its mode: CW, RTTY, USB, LSB, AM, FM
MUTED / vol / pitch / filter	Audio in the browser: mute, volume, CW pitch, filter width; tune radio: a click also moves the radio
Show chips	CQ, TEST, UP, QSO: what is drawn on the spectrum. All / Runners: quick filters
RUN OFF / ON, window, CALLERS	RUN mode (chapter 5)
labels / kept	Full or compact labels; how long a spot stays on screen
Waterfall ON, history, floor, range, Pause	Waterfall on/off (off = less CPU in the browser), its length and its colours
Spots / Signals / Log	Published spots, every signal being decoded, messages of the program
Decoder (bottom)	Real-time decoder of the station you listen to

Labels: green CQ, orange TEST, violet UP. A grey label with a **?** is a call read but not (yet) published: the skimmer is not sure enough. Click a label or a spot to listen to the station (and to tune the radio with **tune radio**).

Whole I/Q: listen anywhere in the band

Whole IQ shows everything the source delivers (about 1.9 MHz with the Icom). Click any signal to listen to it in the browser, in CW, RTTY, USB, LSB, AM or FM, with the filter width you choose. The decoders keep working on the analysed segments whatever you listen to.



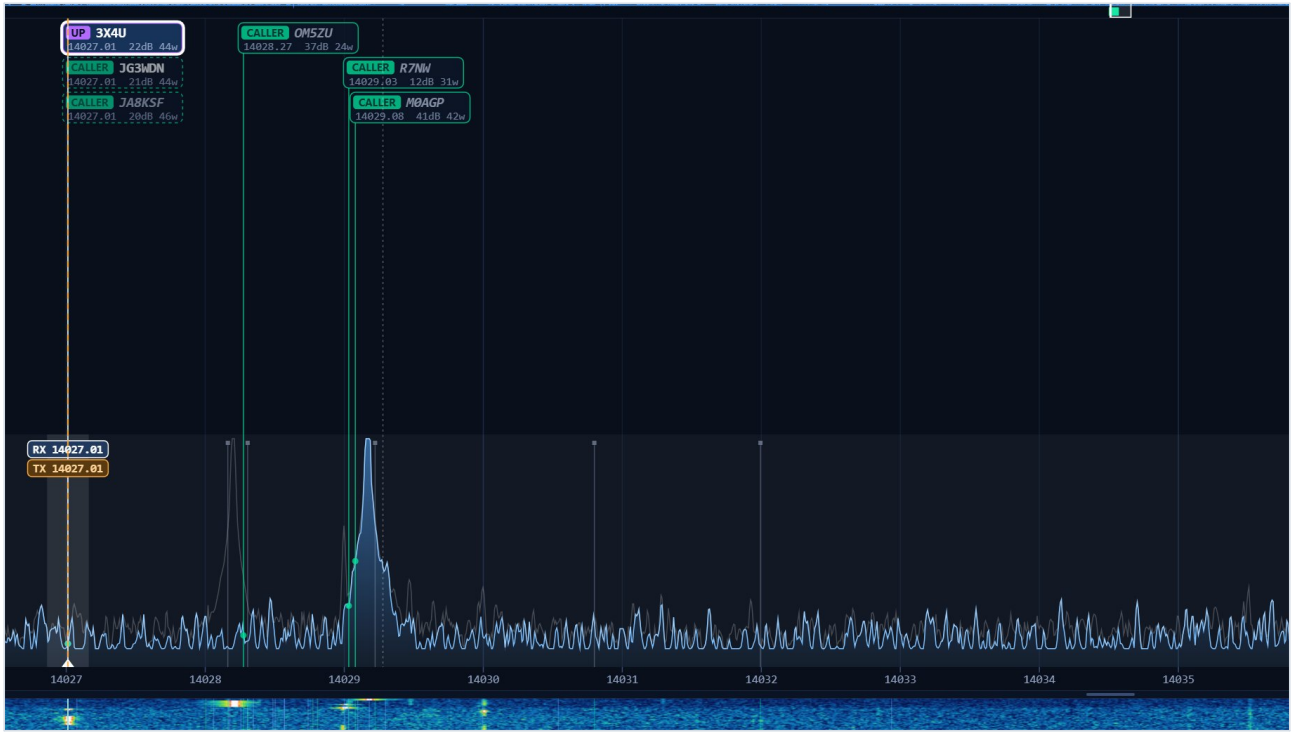
Whole IQ: the full bandwidth of the source; the mode menu lets you listen in CW, RTTY, USB, LSB, AM or FM.

Bands

The analysed ranges are in the `[bands]` section of the INI file (for example `20m = 14000-14070`; several ranges with commas; `off` = not analysed). Changes from the GUI are applied live.
`VKVSkimmer_CW_with_logs.bat contest` uses the wider `[bands.contest]` table instead.

5. RUN mode

RUN mode is for when **you** are running (calling CQ): the skimmer shows the stations that call you around your frequency, a few seconds after they call.



RUN mode: CALLER labels around the run frequency; dashed italic labels are callers read once (provisional). RX and TX markers show the receive and transmit frequencies.

1. Tune the radio (MAIN) to your run frequency.
2. Click **RUN OFF** in the Show row: it becomes **RUN ON**. The RUN window follows the radio.
3. Choose **UP** or **DOWN** and the **window** width: 0 = simplex (+/- 500 Hz), up to 15 kHz for split operation.
4. The spectrum now shows only the stations calling you (**CALLER**); the list shows callers and runners (**CALLERS+RUN**). The CALLERS buttons toggle these views.
5. **Click** a caller: the SUB receiver is tuned to it (dual watch). Your TX frequency on MAIN never moves.
6. **Double click** a caller: the QSO is written to the ADIF file (`VKVSkimmer_CW.adi` or `VKVSkimmer_RTTY.adi`).
7. Click **RUN ON** again to leave RUN mode: the previous view is restored.

> **NOTE**
> A fast second engine analyses only the RUN window (CW: every 1.5 s over 9 s instead of every 5 s over 15 s; RTTY: every 2 s over 8 s), so a caller appears quickly. A call read once is shown at once as provisional (dashed, italic) and becomes normal when the main engine confirms it. The RUN engine never spots: nothing it reads goes to telnet or RBN.

SUB tuning and the window that follows the radio need the Icom direct I/Q input. With an SDR the click moves the listen frequency inside the I/Q.

RUN mode settings in the INI

Key	Default	Meaning
`run_step`	`1.5 (RTTY 2)`	RUN engine: analysis every run_step seconds (GUI only, it never spots)
`run_win`	`9 (RTTY 8)`	RUN engine: analysis window in seconds (9-15)
`adif_file`	`VKVSkimmer_CW.adi / VKVSkimmer_RTTY.adi`	ADIF file of the QSOs logged with a double click on a CALLER
`runner_hold_seconds`	`300`	Seconds a station stays RUNNER after its last CQ / TEST / UP / TU

6. Telnet cluster

The skimmer is its own DX cluster. Every published spot is sent as a standard line, for example:

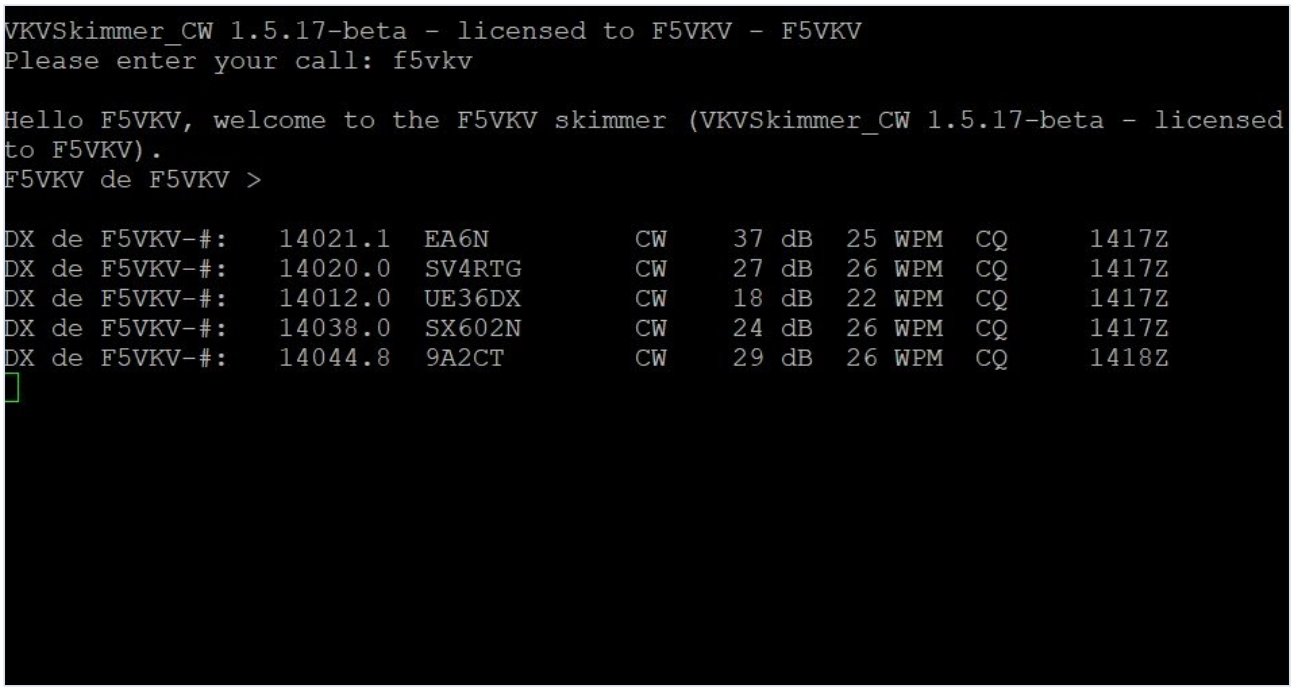
```
DX de F5VKV-#:   14021.1  F5VKV        CW    37 dB  25 WPM  CQ     1417Z
```

Port	CW	RTTY	Content
port	7300	7301	Spots (filtered by spot_mode, default cq: stations heard calling)
heard_port	7310	7311	Every confirmed call, whatever spot_mode. 0 = off

With a telnet client

```
telnet 127.0.0.1 7300
```

Type your callsign when asked. The Windows telnet client must be enabled (Windows Features > Telnet Client); PuTTY works too (connection type Telnet).



A telnet client connected to the skimmer cluster.

N1MM Logger+

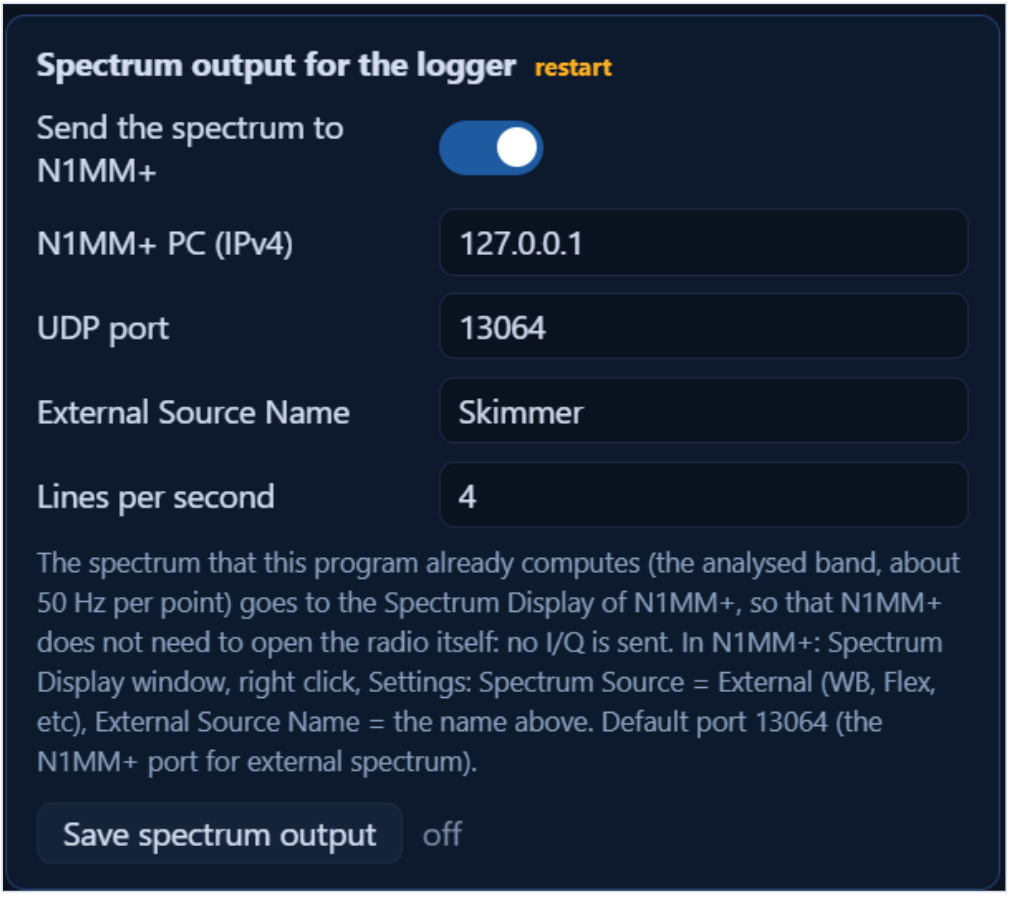
1. Open the Telnet window.
2. In the Clusters tab add a cluster with address `127.0.0.1` and port `7300` (CW) or `7301` (RTTY).
3. Select it and click Connect. The skimmer spots appear in the Telnet window, in the band maps and on the Spectrum Display.

DXLog and other loggers

Add a telnet cluster with host `127.0.0.1` and the same port. From another PC on the network use the IP address of the skimmer PC (the telnet ports accept connections from the network; allow them in the Windows firewall).

7. Spectrum in N1MM+

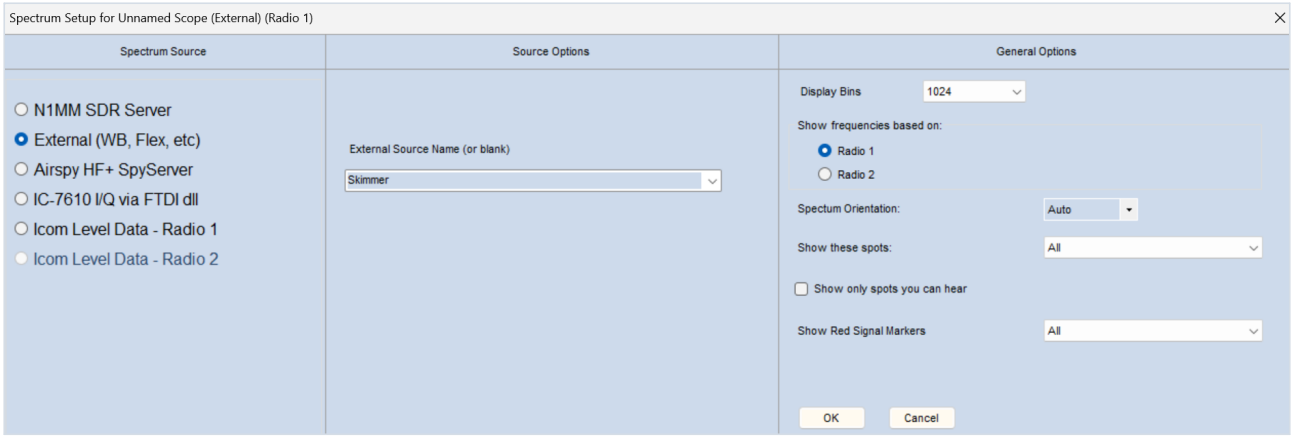
The skimmer can send the spectrum it already computes (the analysed band, about 50 Hz per point) to the N1MM+ Spectrum Display. N1MM+ does not need to open the radio: no I/Q is shared.



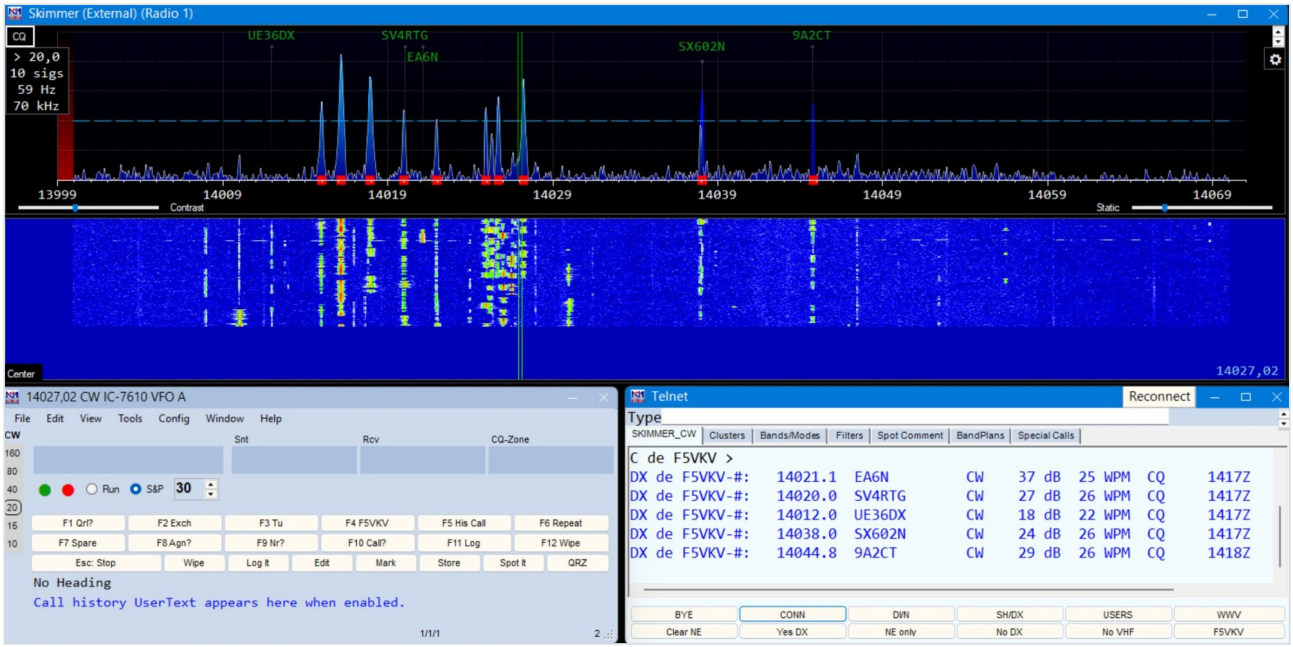
Skimmer: Settings > Spectrum output for the logger.

1. In the skimmer: Settings > **Spectrum output for the logger**: switch on **Send the spectrum to N1MM+**, N1MM+ PC `127.0.0.1`, UDP port `13064`, External Source Name `Skimmer`, lines per second `4`. Click **Save spectrum output** and restart the skimmer.
2. In N1MM+: open the Spectrum Display window, right click, Settings.
3. Spectrum Source: **External (WB, Flex, etc)**; External Source Name: **Skimmer** (the same name as in the skimmer). OK.

> **NOTE**
> The RTTY skimmer uses the name `Skimmer RTTY` by default: use that name in N1MM+ for the RTTY spectrum. The name in N1MM+ must match the one in the skimmer.



N1MM+: Spectrum Setup with the External source named Skimmer.



N1MM+ with the skimmer spectrum and the skimmer cluster in the Telnet window.

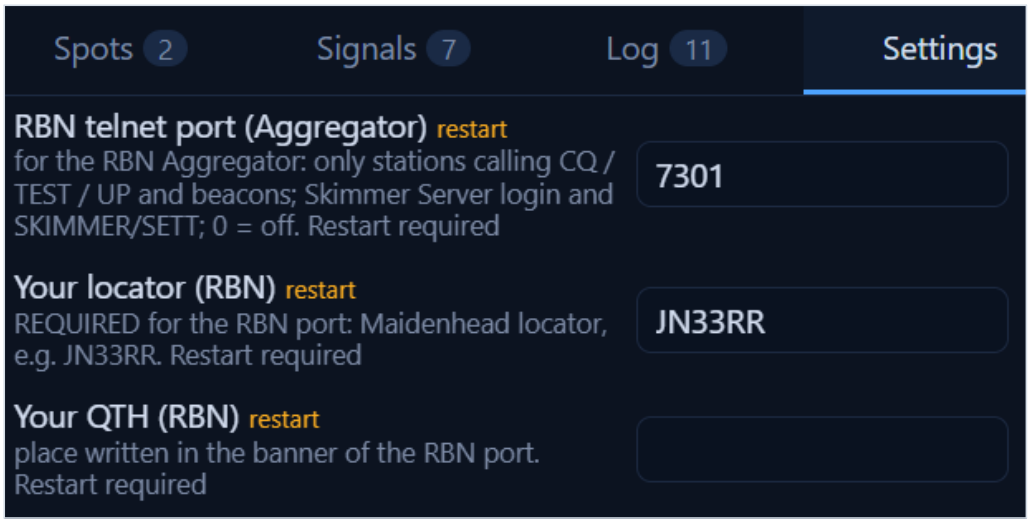
Key	Default	Meaning
`spectrum_out`	`no`	n1mm: send the spectrum to N1MM+ (Spectrum Source External). no: off
`spectrum_host`	`127.0.0.1`	Address that receives the spectrum (the N1MM+ PC)
`spectrum_port`	`13064`	UDP port (the N1MM+ port for external spectrum)
`spectrum_name`	`Skimmer`	Name of this spectrum source in N1MM+
`spectrum_rate`	`4`	Spectrum lines per second (0.5-10)

8. Sending spots to the RBN (RBN Aggregator)

The RBN Aggregator (free, from reversebeacon.net; it needs Microsoft .NET Framework 4) receives the spots of a skimmer over telnet and forwards them to the Reverse Beacon Network. VKVSkimmer has a dedicated port that behaves like VE3NEA's Skimmer Server, so Aggregator connects to it as usual.

REQUIRED
Two settings are **required**: the **RBN telnet port** (`rbn_port`, 0 = off) and **your locator** (`operator_grid`, 4 or 6 characters). Your callsign must be set too. After changing the port or the locator, **restart** the skimmer.

In the skimmer

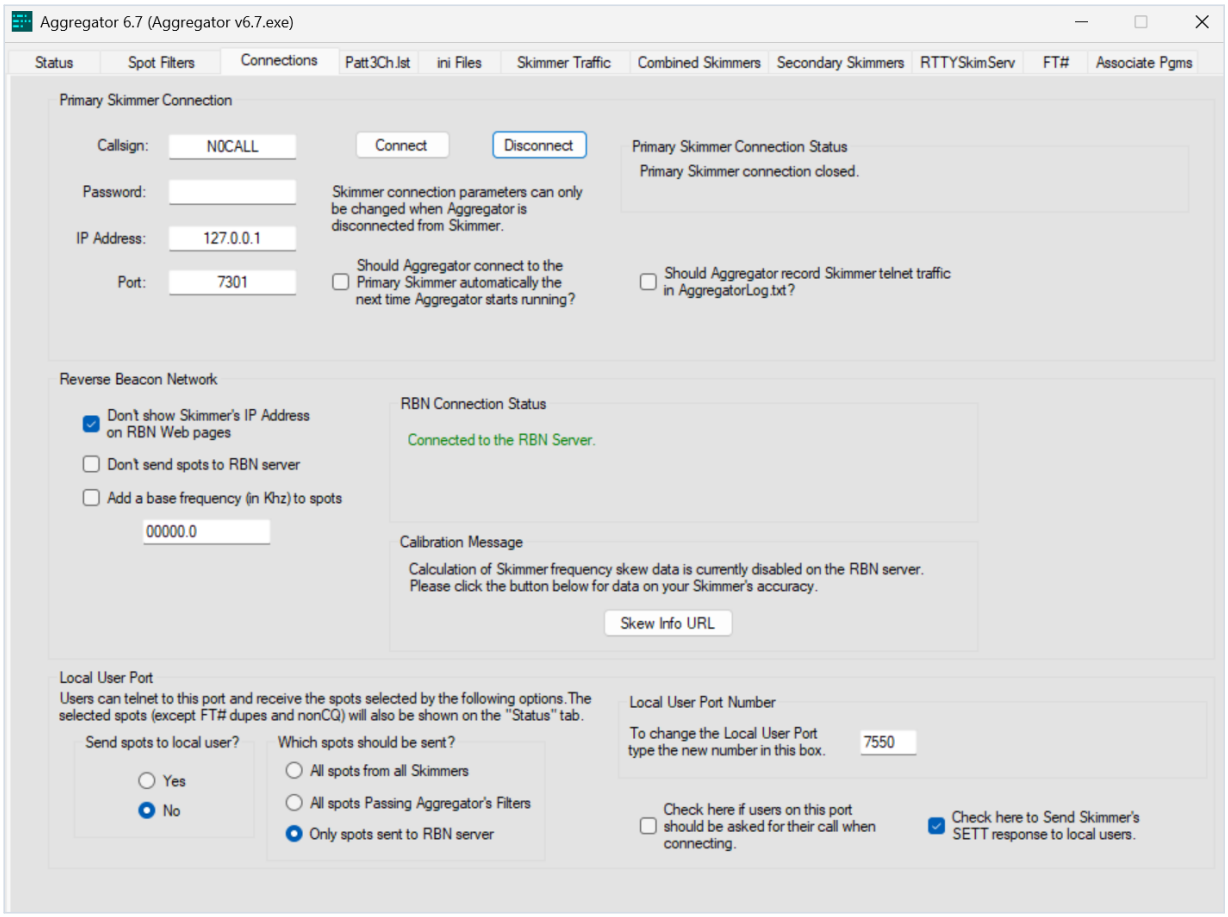


Skimmer: Settings, RBN section (here port 7301, used while the RTTY skimmer is not running: see the note below).

1. Settings > **RBN telnet port (Aggregator)**: a free port, for example `7302`.
2. **Your locator (RBN)**: your Maidenhead locator, for example `JN33RR`.
3. **Your QTH (RBN)**: optional, written in the banner.
4. Restart the skimmer.

> **NOTE**
> Pick a port no other program uses. The RTTY skimmer's cluster uses 7301 by default: if you run both skimmers, give each one its own RBN port (for example 7302 for CW and 7303 for RTTY).

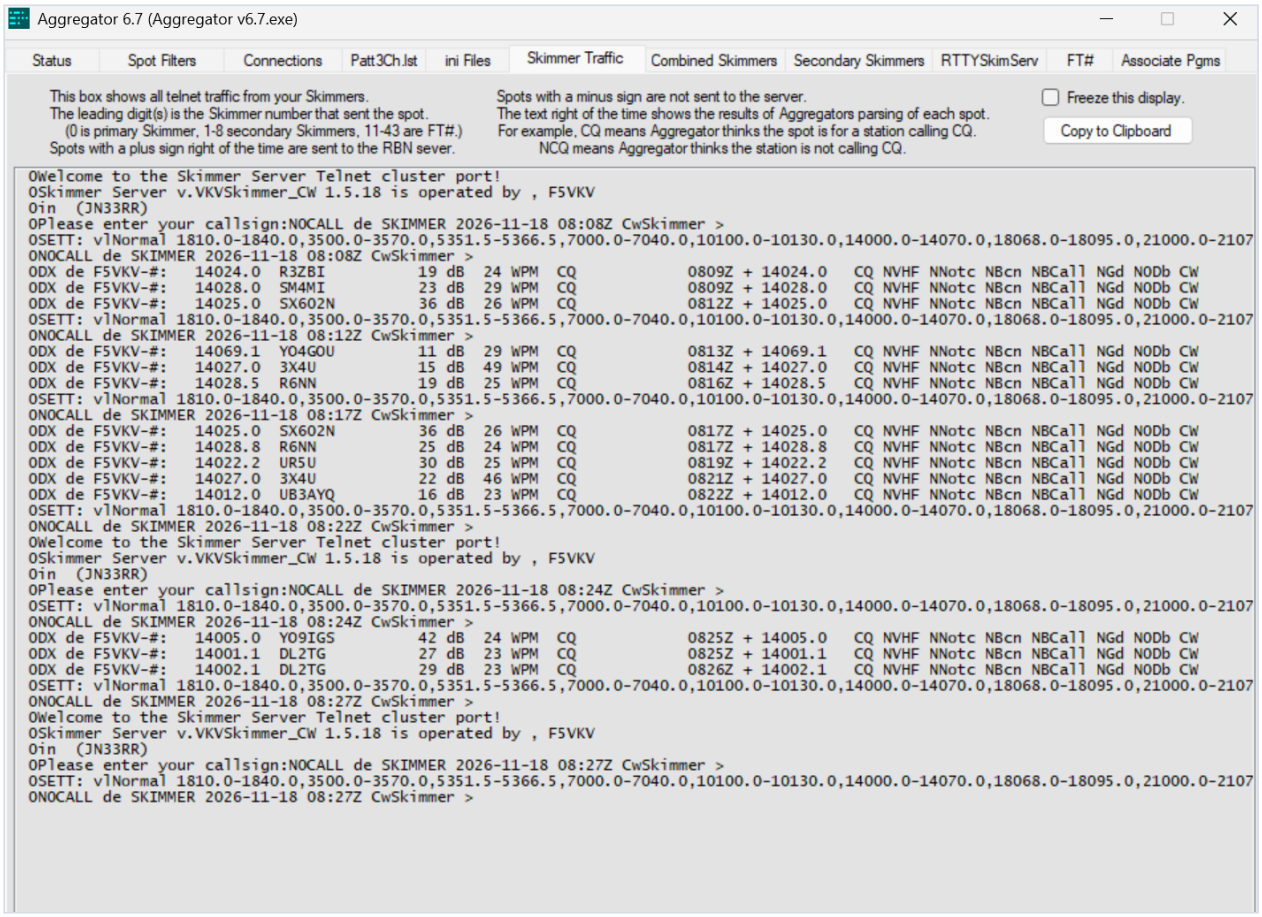
In RBN Aggregator: Connections tab



Aggregator 6.7, Connections tab: primary skimmer on 127.0.0.1 and the skimmer's RBN port.

1. Primary Skimmer Connection: **IP Address** `127.0.0.1` (or the skimmer PC), **Port** = the skimmer's RBN port. Password: empty.
2. Callsign: the name Aggregator uses to log in to the skimmer. The spots always carry the callsign set in VKVSkimmer.
3. Click **Connect**. Tick *Should Aggregator connect to the Primary Skimmer automatically* to reconnect at every start.
4. RBN Connection Status must show **Connected to the RBN Server**.

Check the traffic: Skimmer Traffic tab



Skimmer Traffic: the skimmer banner, the SETT answer with the analysed ranges, and the spots (+ = sent to the RBN).

- The banner shows `Skimmer Server v.VKVSkimmer_CW 1.5.19 is operated by ..., F5VKV` and your locator.
- The `SETT:` line shows the validation level and the analysed ranges.
- Spot lines with **+** after the time are sent to the RBN.
- Only stations calling CQ / TEST / UP and beacons are sent; TEST and UP are sent as CQ, as Skimmer Server does (`rbn_comment = cq`).

To feed both skimmers through one Aggregator, connect the second one in Aggregator's **Secondary Skimmers** tab (see the Aggregator user guide on reversebeacon.net).

Key	Default	Meaning
`rbn_port`	`0`	Telnet port for the RBN Aggregator (Skimmer Server compatible). 0 disables it
`operator_grid`	`(empty)`	Your Maidenhead locator: required by rbn_port
`operator_qth`	`(empty)`	Your QTH, shown in the banner
`operator_name`	`(empty)`	Your name, shown in the banner
`rbn_validation`	`vlNormal`	Validation level in the SETT answer, used with patt3ch.lst
`rbn_comment`	`cq`	cq: TEST and UP are sent as CQ; keep: sent as they are
`rbn_prompt`	`CwSkimmer / RttySkimmer`	Word in the prompt of the RBN port
`patt_file`	`patt3ch.lst`	RBN callsign patterns used for the confirmations

9. RBN Reference (incoming RBN spots)

RBN Reference is the opposite direction: the skimmer reads the RBN spots and uses them to **confirm**, **hold** or **reject** its own candidates. It never creates or corrects a callsign. Enable it with `rbn_reference = yes` in the INI file; it logs in to the RBN with your callsign. If the RBN cannot be reached, the skimmer works normally on its own evidence.

- A call heard by 2 or more RBN skimmers on the same frequency is published at once.
- A truncated or busted reading, when the RBN strongly confirms a similar call there, is rejected (F5VK when the RBN hears F5VKV).
- A call never heard by the RBN, on a frequency where many skimmers hear another station, is a caller of that station: not published.
- A call without RBN evidence waits up to `rbn_reference_hold_sec` (default 5 s), then the local decision applies.
- Inside the RUN window of the GUI the calls are not checked against the RBN.

Key	Default	Meaning
`rbn_reference`	`no`	yes: every publishable call is checked against the RBN spots
`rbn_reference_host`	`telnet.reversebeacon.net`	RBN telnet server
`rbn_reference_port`	`7000`	RBN telnet port (CW and RTTY)
`rbn_reference_hold_sec`	`5`	Seconds a call waits for RBN evidence (0-15)

10. Contests

- `spot_mode` (also in the GUI): **cq** (default: stations heard calling in the last 3 minutes), runner, runner_unknown or all.
- `contests.txt` lists the contests and the words they send after CQ (CQ WPX, CQ RDXC...). Download the new copy from vkvskimmer.com and replace the file; nothing is downloaded automatically.
- `calling_words`: extra words that mean calling, comma separated.
- `VKVSkimmer_CW_with_logs.bat contest` / `VKVSkimmer_RTTY_with_logs.bat contest`: the wider `[bands.contest]` ranges.

11. Troubleshooting

Problem	What to check
Stations in the GUI but no spots on telnet	Your callsign (red SET YOUR CALLSIGN warning). spot_mode = cq spots only stations heard calling.
Aggregator does not connect	rbn_port not 0, operator_grid set, skimmer restarted, same port in Aggregator, port not used by another program.
Logger does not connect	Port 7300 (CW) / 7301 (RTTY); another program on the same port; Windows firewall when the logger is on another PC.
Icom: no input	USB I/Q Package installed, IC-7610 firmware 1.20+, [USB 2] on a USB 3.0 port with a USB 3.0 cable, FTD3XX.dll next to the program.
RTL-SDR: no input	rtl_tcp.exe and its DLLs next to the program, WinUSB on Interface 0 with Zadig (again after a big Windows update), dongle not used by another program.
SDRplay: no input	SDRplay API 3 installed and its service running; SDRuno / SDRconnect closed.
Lost data / slow analysis	Settings > Profiles: LIGHT. Laptop on AC power. Waterfall OFF in the browser.
GUI does not open	Open http://127.0.0.1:8073/ (CW) or :8074/ (RTTY) by hand; web_open = yes.
Some stations are never spotted	By design: if a call, a prefix or the CQ role is uncertain, it is not published.

Reporting a problem

Write to the address on vkvskimmer.com with your callsign, radio / input, Windows version, UTC time, frequency, the call you expected, and the log ZIP that the `_with_logs.bat` file writes in the `logs` folder when the program stops (start the program with that BAT file when you want to report a problem).

Appendix A. All settings of skimmer.ini (CW)

Every key belongs to `[general]`. A line starting with `;` is not active: the program uses the value shown (its default). Remove the `;` to change it. Settings changed in the GUI are written into the same lines (a copy of the old file is kept as .bak). Command-line options override the file.

YOUR STATION

Key	Default	Meaning
`callsign`	`(empty)`	Your callsign: the spotter of every line (DX de CALL-#), the telnet greeting, the RBN Aggregator banner and the RBN login. REQUIRED: without it the skimmer decodes but publishes nothing. Also in the GUI (Settings > Your callsign), applied at once. Portable: F/F5VKV (the prefix must exist in cty.dat) or F5VKV/P.

TELNET OUTPUT (logger, cluster, RBN Aggregator)

Key	Default	Meaning
`port`	`7300`	Telnet port of the spots for the logging program (N1MM+, DXLog...).
`heard_port`	`7310`	Telnet port of the HEARD stream: every confirmed call, whatever spot_mode. 0 disables it.
`spot_mode`	`cq`	Which stations are spotted: cq (heard calling in the last 3 min), runner, runner_unknown or all. Also in the GUI.
`calling_words`	`SEQP`	Extra contest words that mean 'calling' (besides TEST and contests.txt). Comma separated.
`contests_file`	`contests.txt`	Contest catalogue: the CQ words of each contest. Maintained by hand, never downloaded.
`spot_ttl_minutes`	`30`	Minutes a spot stays in the GUI lists and on the scope. 0 keeps it until CLEAR.
`rbn_port`	`0`	Telnet port for the RBN Aggregator (Skimmer Server compatible). 0 disables it. Needs operator_grid.
`operator_grid`	`(empty)`	Your Maidenhead locator (4 or 6 characters, e.g. JN33RR): required by rbn_port.
`operator_qth`	`(empty)`	Your QTH, shown in the banner of the RBN port.
`operator_name`	`(empty)`	Your name in the banner of the RBN port.
`rbn_server_name`	`VKVSkimmer_CW`	Program name shown in the banner of the RBN port.
`rbn_server_version`	`(empty)`	Version shown in the banner of the RBN port. Empty: the version of this program.
`rbn_prompt`	`CwSkimmer`	Word in the prompt of the RBN port (the real CW Skimmer uses CwSkimmer; the RTTY skimmer uses RttySkimmer).
`rbn_validation`	`vlNormal`	Validation level sent in the SKIMMER/SETT answer and used with patt3ch.lst: vlMinimal, vlNormal, vlAggressive.
`rbn_comment`	`cq`	Comment sent to the RBN: cq (TEST and UP are sent as CQ, as the real Skimmer Server) or keep.
`patt_file`	`patt3ch.lst`	RBN callsign patterns used for the confirmations. off: not used.

RBN REFERENCE (incoming RBN spots used to validate our own spots)

Key	Default	Meaning
rbn_reference	no	yes: every publishable call is checked against the spots of the other RBN skimmers. Inside the RUN window it is not checked.
rbn_reference_host	telnet.reversebeacon.net	RBN telnet server used for the reference.
rbn_reference_port	7000	Port of the RBN telnet server (7000: CW and RTTY).
rbn_reference_hold_sec	5	Seconds a call without RBN confirmation waits for RBN evidence before the local decision (0-15).

BROWSER GUI

Key	Default	Meaning
web	yes	yes: start the browser GUI. no: no GUI.
web_open	yes	yes: open the GUI in the browser at start-up.
web_port	8073	HTTP port of the browser GUI.
web_lan	no	yes: the GUI is reachable from the local network (there is no password).
rx_gap_spaces	no	Real-time decoder of the station you listen to: word spaces from the pauses too (yes) or only from the model (no).

RADIO AND INPUT SOURCE

Key	Default	Meaning
input	icom	Input: icom (IC-7610 / IC-7760 USB I/Q + CI-V), sound_iq, sound_audio, sound_if, rtlsdr or sdrplay. Easier: Settings > Input source.
radio	auto	Radio model: ic7610, ic7760 or auto (the first radio found).
sub	no	yes: use the I/Q of the SUB receiver instead of MAIN.
conj	no	yes: the I/Q spectrum is inverted (spot frequencies mirrored around the centre).
cal	0	Frequency correction added to every spot, in Hz.
freq_poll_ms	100	How often the radio frequency is read, in ms (30-1000).
retune_guard_ms	100	I/Q around a retune of the radio that is not used, in ms (the radio moves before the program knows). 0: off.
tune_mode	no	yes: a click on a station also sets the receiver to CW when it is in another mode (CW-R is left alone).
sound_device	(empty)	Sound card inputs: name of the Windows recording device (a part of it is enough).
sound_channel	left	sound_audio: left or right channel.
sound_pitch	700	sound_audio: the CW pitch, the audio tone that is on the dial frequency (Hz).
sound_invert	no	sound_audio: yes for CW-R / LSB (a higher station gives a lower tone).
sound_rf_khz	14025	sound_audio and sound_if: the dial frequency of the radio in kHz (no CAT: change it when you change frequency).

Key	Default	Meaning
sound_center_khz	(empty)	sound_iq: the frequency at the centre of the I/Q, in kHz.
sound_rate	0	sound_iq: sample rate. 0: the rate Windows has for the device.
sound_swap_iq	no	sound_iq: yes when Left is Q and Right is I (spectrum reversed).
sound_exclusive	auto	sound_iq and sound_if: auto opens the device in exclusive mode (bit-exact, no Windows mixer); no: shared mode.
if_center_hz	12000	sound_if: IF centre in the audio stream in Hz (Icom: 12 kHz at 48 kHz).
if_half_bw_hz	7500	sound_if: half of the analysed width around the dial, in Hz (max about 10000 at 48 kHz).
if_offset_hz	0	sound_if: correction of the IF centre against the dial, in Hz.
if_invert	no	sound_if: yes when the spectrum comes out reversed.
sdr_center_khz	(empty)	SDR: centre used at the last run, in kHz (written by the program; empty: 20 m).
sdr_rate	2048000	RTL-SDR: sample rate (1024000, 1400000, 1800000, 2048000, 2400000).
rtl_direct	auto	RTL-SDR: direct sampling: auto (Q branch below 24 MHz), q, i, off.
rtl_agc	yes	RTL-SDR: digital AGC of the RTL2832 (recommended in direct sampling).
rtl_gain	auto	RTL-SDR: tuner gain in dB or auto (no effect in direct sampling).
rtl_bias	no	RTL-SDR: bias tee (only for an LNA / active antenna fed through the coax).
rtl_tcp_port	1234	RTL-SDR: port of rtl_tcp on 127.0.0.1.
rtl_tcp_path	(empty)	RTL-SDR: path of rtl_tcp.exe. Empty: the one next to the program.
rtl_ppm	0	RTL-SDR: frequency correction in ppm.
rtl_index	0	RTL-SDR: which dongle (0: the first).
sdrplay_gr	40	SDRplay: IF gain reduction, 20-59 dB.
sdrplay_lna	3	SDRplay: LNA state (0: most gain).
sdrplay_agc	no	SDRplay: IF AGC.
sdrplay_bias	no	SDRplay: bias tee (RSP1A, RSP1B, RSP2, RSPduo, RSPdx).
sdrplay_serial	(empty)	SDRplay: serial number. Empty: the first RSP.

SPECTRUM OUTPUT FOR N1MM+

Key	Default	Meaning
spectrum_out	no	n1mm: send the spectrum of the analysed band to N1MM+ (Spectrum Source External). no: off.
spectrum_host	127.0.0.1	Address that receives the spectrum.
spectrum_port	13064	UDP port of the spectrum.
spectrum_name	Skimmer	Name of this spectrum source in N1MM+.
spectrum_rate	4	Spectrum lines per second (0.5-10).

DECODER PROFILE (CW)

Key	Default	Meaning
cpu_profile	auto	CPU profile: auto (adapts to the load), light (low CPU) or maximum (fast PC). Also in the GUI.
decoder_mode	profile	profile: use cpu_profile. Advanced / benchmark: DEEPCW, DEEPCWN, NORMAL, STRONG, AGGRESSIVE, LUDICROUS.
engine	hybrid	Decoder engine (with decoder_mode profile it is set by the profile): viterbi, classic, both, deep, hybrid.
deep_all	yes	hybrid: DeepCW on every signal (yes) or only where viterbi found no known callsign (no).
deep_max	80	At most this many DeepCW decodes per analysis step and segment.
deep_adapt	yes	yes: lower the DeepCW budget of a segment by itself when its analysis gets too slow.
decoder_priority	deep	Who reads first: deep (DeepCW primary, viterbi as extra evidence) or viterbi (the fusion of 1.5.9).
stable_skip	yes	Under load, a station already spotted and read the same is decoded every other step.
cw_neural	e04	Neural network: e04 (ONNX, needs onnxruntime.dll) or ve3nea (deepcw.bin).
e04_dll	onnxruntime.dll	onnxruntime.dll. Empty: the one next to the program.
e04_cw_model	models\e04_deepcw_model_narrow.onnx	Narrow DeepCW model of e04 (its .json must have the same name + .json).
e04_cw_rx_model	models\e04_deepcw_model.onnx	Standard DeepCW model (real-time decoder of the GUI and the standard reading). Missing: the narrow one is used.
e04_threads	1	Threads of ONNX Runtime per decode.
deepcw	deepcw.bin	DeepCW weights of VE3NEA (cw_neural ve3nea).
deep_minconf	0.75	A call found only by the neural network and not in MASTER.SCP needs this mean character confidence.
standard_max_per_step	2	STRONG: signals per step with a second reading. AGGRESSIVE: signals per step with the standard model.
standard_bandwidth_hz	250	Band around the tone read by the standard model, in Hz (100-800).
standard_min_snr	3	STRONG: no second reading under this excess over the noise, in dB.
decoder_trace	no	Benchmark: one CSV line per signal (texts of the decoders, calls, ms). no: off.
decoder_stats	yes	[decoder] line in the status file every 10 s and a summary at the end.

CPU AND THREADS

Key	Default	Meaning
maxseg	auto	Width of a segment in kHz (one thread each). auto: the smallest piece of 20-35 kHz that fits the CPUs of the PC.
decode_threads	auto	Threads shared by all the segments for the decoding. auto: logical CPUs - 1 (max 8); 0: none.

Key	Default	Meaning
cores	0	Logical CPUs to assume (0: ask the system). Used by maxseg auto and the [status] line.
priority	normal	Priority of the analysis threads: normal or below (below: the logger and other programs come first).
analysis_deadline	0.70	An analysis step stops decoding more signals after this fraction of the step (strongest first). 0: no limit.

SIGNAL DETECTION AND CALLSIGN RULES (CW)

Key	Default	Meaning
hits	2	Confirmations (analysis windows) needed before a spot.
maxwpm	50	Decodes faster than this (WPM) are ignored.
dyn	12	Keying detection threshold (dynamic range), in dB.
exc	12	Keying detection threshold (excess over the noise), in dB.
win	15	Analysis window in seconds.
step	5	Analysis step in seconds.
impulse_blanker	auto	Impulse noise blanker on the I/Q (isolated broadband spikes): auto / yes, or no.
multi_peak	yes	yes: next to a signal a second, weaker peak can be followed too (two stations close together).
adaptive_filter	no	Extraction filter of each signal: no (as 1.4.6), narrow (yes) or tone (reads twice).
strong_level_db	8	Viterbi decision level on very strong signals: at most this many dB below the peak (stops false dots / dashes). 0: off.
octave_guard	yes	Speed (octave) guard of viterbi: yes, fast or no.
agreed_no_hold	yes	yes: no spot_hold_sec wait when every confirmation was read whole by both viterbi and DeepCW.
spot_hold_sec	10	Seconds the first spot of a risky reading waits for more evidence.
sic	no	Experimental: subtract a strong neighbour and try viterbi again on the weaker signal.
pair_min_hz	75	Minimum distance of the second peak from the main one, in Hz.
pair_valley_db	4	Valley needed between the two peaks, in dB.
pair_persist	3	Analysis windows the second peak must persist before it is decoded.
unstable_filter	yes	Hold back a call not in MASTER.SCP while 2+ near variants are read on the same frequency.
runner_hold_seconds	300	Seconds a station stays RUNNER after its last CQ / TEST / UP / TU call.

RUN MODE OF THE GUI

Key	Default	Meaning
run_step	1.5	RUN window engine: analysis every run_step seconds (GUI only, it never spots).
run_win	9	RUN window engine: analysis window in seconds (9-15).

Key	Default	Meaning
adif_file	VKVSkimmer_CW.adi	ADIF file of the QSOs logged with a double click on a CALLER.

FILES AND UPDATES

Key	Default	Meaning
scp	MASTER.SCP	Known callsigns (more weight to known calls, repair of one unreadable letter). Strongly recommended.
cty	cty_wt.dat	Country file: calls with no DXCC prefix are rejected, the country is shown.
auto_update	yes	yes: at start-up, at most once every update_days, download MASTER.SCP and cty_wt.dat if they changed.
update_days	1	Days between two update checks.
scp_url	https://www.supercheckpartial.com/MASTER.SCP	Download address of MASTER.SCP.
cty_url	https://www.country-files.com/cty/cty_wt.dat	Download address of cty_wt.dat.

Bands

[bands]: analysed ranges in kHz, several ranges separated by commas, off = band not analysed. Changes made in the GUI are applied live. [bands.contest]: optional wider ranges, used with VKVSkimmer_CW_with_logs.bat contest.

Band	[bands]	[bands.contest]
160m	1810-1840	1810-1840
80m	3500-3570	3500-3570,3576-3600
60m	5351.5-5366.5	-
40m	7000-7040	7000-7070,7076-7100
30m	10100-10130	-
20m	14000-14070	14000-14072,14080-14165
17m	18068-18095	-
15m	21000-21070	21000-21072,21080-21165
12m	24890-24915	-
10m	28000-28070	28000-28072,28080-28165
6m	50000-50100	-
2m	144000-144170	-
70cm	432000-432100	-
23cm	1296000-1296150	-

Appendix B. rtty.ini: what is different

rtty.ini has the same structure and most of the same keys as skimmer.ini. These are the keys that exist only in rtty.ini and the defaults that differ.

Keys only in rtty.ini

Key	Default	Meaning
tune_to	mark	A click on a station tunes the radio to its mark tone (mark, the RTTY convention) or to the centre between the tones (centre).
vfo_segment	yes	yes: when the radio is outside the band table, a segment around its frequency is analysed too.
vfo_width	15	Width of that segment around the radio frequency, in kHz (5-60).
baud	45.45	Baud rate: 45.45 (amateur standard), 50 or 75. DeepRTTY works at 45.45 only.
shift	170	Shift in Hz.
polarity	auto	Polarity: auto (both tried), normal (mark is the higher frequency) or reverse.
usos	yes	yes: unshift on space (Baudot back to letters after a space).
decoder	soft	Conventional decoder: soft (default) or classic.
rtty_neural	e04	Neural network: e04 (DeepRTTY, needs onnxruntime.dll) or classic (conventional decoder only).
e04_rtty_model	models\e04_deeprtty_model.onnx	DeepRTTY model (its .json must have the same name + .json).
e04_rtty_max	16	At most this many DeepRTTY decodes per analysis step and segment (strongest tone pairs first).
e04_rtty_minconf	0.80	Mean character probability needed to use the DeepRTTY text.
min_snr	8	Weaker tone of the pair above the local noise, in dB.

Different defaults

Key	skimmer.ini (CW)	rtty.ini (RTTY)
port	7300	7301
heard_port	7310	7311
rbn_server_name	VKVSkimmer_CW	VKVSkimmer_RTTY
rbn_prompt	CwSkimmer	RttySkimmer
web_port	8073	8074
sound_pitch	700	2125
sound_invert	no	yes
sound_rf_khz	14025	14085
spectrum_name	Skimmer	Skimmer RTTY
maxseg	auto	30
win	15	12
runner_hold_seconds	300	900

Key	skimmer.ini (CW)	rtty.ini (RTTY)
run_step	1.5	2
run_win	9	8
adif_file	VKVSkimmer_CW.adi	VKVSkimmer_RTTY.adi

RTTY bands

Band	[bands]	[bands.contest]
160m	1838-1842	-
80m	3580-3600	3570-3650
40m	7035-7050	7025-7100
30m	10130-10150	-
20m	14080-14100	14070-14150
17m	18100-18110	-
15m	21080-21120	21070-21150
12m	24910-24930	-
10m	28080-28120	28070-28150
6m	50300-51000	-

Credits: DeepCW by Alex Shovkoplyas, VE3NEA (MIT); DeepCW / DeepRTTY models by e04 (MIT); ONNX Runtime (Microsoft, MIT); cty.dat / cty_wt.dat by Jim Reisert, AD1C (MIT); Super Check Partial (MASTER.SCP); Reverse Beacon Network. Full licence texts: THIRD_PARTY_NOTICES.txt and the LICENSES folder of the package.