

# **RTTY Skimmer Server**

**v.1.0**

## **User's Manual**

<http://www.dxatlas.com/RttySkimServ>

**Copyright © 2014 Afreet Software, Inc.**

# CONTENTS

CONTENTS.....	2
OVERVIEW.....	3
Computer.....	4
Operating System.....	4
SDR Receiver.....	4
USER INTERFACE.....	7
Status Window .....	7
Band Scope .....	8
CONFIGURATION FILES .....	10
RttySkimserv.ini .....	10
Contests.ini.....	14
TELNET COMMANDS .....	16
OTHER FEATURES .....	17
Additional Filtering Options .....	17
Additional Files.....	18
TROUBLESHOOTING.....	19
REFERENCES .....	20

## OVERVIEW

RTTY Skimmer Server is a Windows program that decodes large numbers of RTTY signals on multiple HF bands simultaneously. The callsigns, frequencies, and signal strength of the decoded stations are available as DX spots on the built-in Telnet cluster server. These spots may be fed to other software, such as contest or general-purpose logging programs, or posted to the Reverse Beacon Network [6] using the RBN Aggregator software.

RTTY Skimmer Server spots the stations received using your own radio. Cluster spots may not be workable, skimmer spots should be.

Another important advantage of RTTY Skimmer Server is that it spots all stations that it can decode, while the stations spotted manually on a DX cluster are only those that someone considered worth spotting. The appearance of stations on the air is typically reported before a traditional DX spot is made by a human operator.

# SYSTEM REQUIREMENTS

## ***Computer***

RTTY Skimmer Server is designed to simultaneously decode hundreds of RTTY signals. This is a CPU-intensive task that requires a fast computer. The faster your machine, the more RTTY signals will be decoded in parallel, and the wider the bandwidths that can be covered. On a Core i7 system, RTTY Skimmer Server can run 400-500 decoders at the same time on 7 x 192-kHz bands. Reducing the per-band bandwidth or the number of bands covered, or both, can help manage CPU usage.

The SSE3 [1] command set must be supported by the CPU. This command set was introduced 10 years ago in Pentium 4, and now virtually all computers have it. Systems that do not support SSE3 are not fast enough to run RTTY Skimmer Server anyway.

At least 8 GB RAM is recommended. RTTY Skimmer Server consumes about 1Gb, the rest is required for the operating system and other applications.

## ***Operating System***

RTTY Skimmer Server was designed to work on Windows 7, 32-bit or 64-bit. It has been tested on Windows 8 and 8.1 machines without incident, but this performance cannot be guaranteed.

## ***SDR Receiver***

QS1R [2] is the only radio directly supported by RTTY Skimmer Server. It offers the capability of monitoring up to 7 bands at up to 192 KHz per band,.

Some other radios are supported via 3-rd party drivers. At the time of this writing, the following drivers were available:

<b>Driver</b>	<b>Supported Radios</b>	<b>Author</b>
HermesIntf [3]	Hermes Angelia Griffin Metis N1GP RTL dongle Afedri	Vasiliy K3IT
Skimmer Listener [4]	USRP	Petr OL5Q
SDR-Bridge [5]	Flex 6000 series	Ed W2RF

The third party driver DLL's should be placed in the program folder where RTTY Skimmer Server is installed. The program loads all of the DLL's on startup. Use the SKIMMER/SDR command as described in the Configuration Files section to get a list of loaded drivers.

## **INSTALLATION AND UNINSTALLATION**

To install RTTY Skimmer Server, run Setup.exe in the download and follow the on-screen instructions. Accept the default directory and other settings when asked. To uninstall the program, open Control Panel, click on Programs and Features (Windows 7), right-click on RTTY Skimmer Server, and click on Uninstall.

## USER INTERFACE

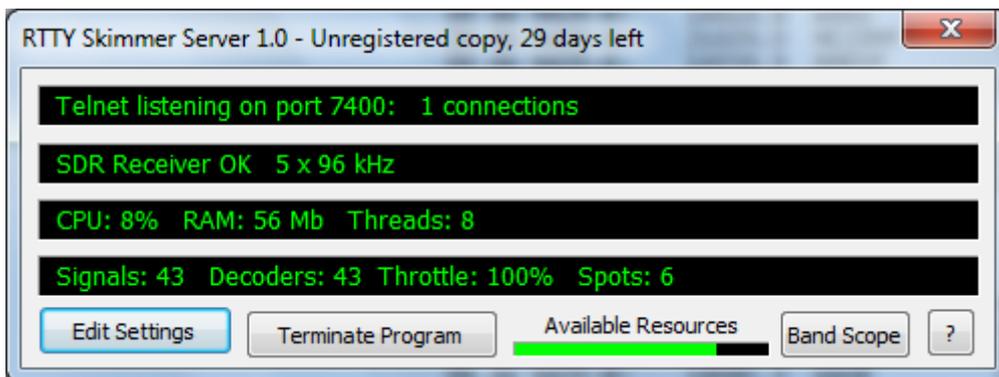
Since RTTY Skimmer Server is a server, it has only the minimum user interface required for its operation. Normally the only visible indicator of RTTY Skimmer Server running on your system is a rectangular yellow icon in the System Tray, as shown on the picture below. Click on the icon to open the Status Window. If you see either a yellow or red marking in a lower corner of the icon, there is an error of some sort which must be corrected.



*Fig.1. The icon in the System Tray*

### **Status Window**

The Status Window has four panels that show the status of the built-in Telnet server and SDR radio, system resources and signal decoding statistics.



The four panels each show the status of an important part of the program. The Telnet panel tells you on what port the Telnet server is listening, and how many users are connected. The SDR Receiver panel tells you the state of the connection between the program and the radio, the number of bands and the bandwidth per band. Yellow type means that the program is in the process of connecting to the radio, while red type means that there is something preventing connection to the radio. In the latter case, the panel displays the error message returned by receiver's driver. In most cases, these are caused by some problem with the hardware, such as a firmware

load from a previous use. This can be remedied by briefly interrupting power to the receiver. After power is resumed, RTTY Skimmer Server will reload the correct firmware.

In the CPU panel, yellow means the CPU load is over 70 percent, while red means it is over 90 percent. In the Telnet panel, red means that there is something wrong in your Telnet settings. In most cases, a red message appears because the specified TCP port number is already in use by some other program.

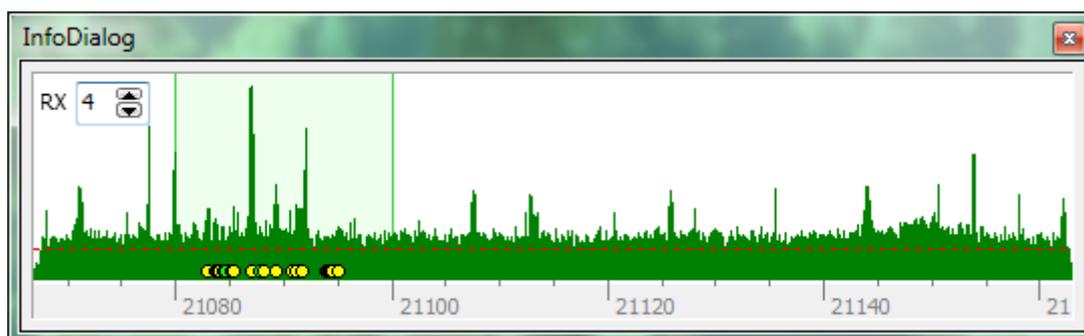
Yellow in the Channel Info panel means that throttling is in effect. This warrants explanation. Throttling occurs whenever the program's input buffer is full – that is, when more signals are being presented for decoding than it can handle, given your computer's CPU capacity. In that situation, the program automatically and randomly reduces the number of signals being decoded until CPU use returns within normal limits.

In addition to the messages in the status panels, some extra detail is available on the mouse-over hints. If you place your mouse cursor over the Telnet panel, you'll see on the mouse-over hint information about any connected Telnet clients. If the mouse cursor is over the Signals panel, you'll see a band-by-band breakdown of the number of signals being decoded.

The Edit Settings button allows you to open the RttySkimServ.ini file (see [Configuration Files](#) below) in your default text editor. Once the editor is closed, the server automatically restarts with the new settings.

Clicking on the red [x] button in Status Window closes the window but does not stop RTTY SkimServer. To stop the program, click on the Terminate Program button, or right-click on the icon in the System Tray and click on Exit in the pop-up menu.

## ***Band Scope***



Click on the BandScope button, and you will be shown a spectrum of the covered portion of each band you are receiving. The RTTY segments (set in your ini file, as explained below) are shown in light green, as above (21080-21100 KHz, in this example). Click the up and down buttons next to the “RX” legend to step up and down through the covered bands. Band 0 is the lowest-frequency band covered.

A yellow dot represents a signal that RTTY Skimmer Server has detected but whose callsign is not yet known. These may not be RTTY signals at all. A green dot represents an RTTY station whose callsign has been decoded. Such stations may be CQ-ing, calling someone, or in a QSO. Depending on the validation setting and whether the program is set only to spot CQ-ing stations, some of these stations may not be spotted.

### ***About Dialog***

This dialog opens when you click on the [?] button in the Status window. It displays the program version, web site address, author’s contact information, and has the Register button that allows you to enter your shareware registration key if you decide to keep the program after the free trial period.

## CONFIGURATION FILES

### ***RttySkimServ.ini***

RTTY Skimmer Server's configuration is controlled by the contents of the RttySkimServ.ini file. To find this file, click on Start (Windows 7) and enter the following in the search box:

***%appdata%\Afreet\Products\RttySkimServ***

When you press Enter, Windows automatically replaces %appdata% with the path to the application data folder.

If RTTY Skimmer Server is running, you can also edit the ini file and automatically restart the program by clicking the Edit Settings button.

Here is a section by section breakdown. In each of the examples that follow, the values shown are those provided by the author when the program is first installed. A semi-colon (;) at the beginning of a line means that the line is a comment or label, and is ignored by the program.

```
[User]
Call=NONE
Name=Unknown
QTH=Unknown QTH
Square=Unknown Gridsquare
```

Enter your callsign, name, QTH (in general, such as state, province, or department), and 6-character Maidenhead grid square. Be sure to enter correct values in this section, these data are required for correct operation of the built-in Telnet server.

```
[SDR]
DeviceName=01 QuickSilver QS1R
Rate=1
FreqCalibration=1.0
```

If you are using a radio other than QS1R, you must change the value of DeviceName and enter its name exactly as it is defined in its driver DLL. To list all valid names, use the SKIMMER/SDR Telnet command as described in the [Telnet Commands](#) section.

Rate determines the bandwidth to be covered, per band. Enter 0 for 48 KHz, 1 for 96 KHz, and 2 for 192 KHz.

FreqCalibration is an adjustment factor to bring your SDR exactly on frequency. The easiest way to do this is to watch the Reverse Beacon Network [6] for spots of a station that you also spot. Divide your spot's frequency by the frequency reported by the consensus of RBN stations spotting the same callsign. The result is the adjustment factor.

```
[Telnet]
Port=7300
PasswordRequired=0
Password=
CqOnly=0
AllowAnn=1
AnnUserOnly=0
AnnUser=
ValidationLevel=1
```

This section determines the settings for the built-in Telnet server. The port can be set to anything above port 1000 so long as it does not conflict with ports in use by other programs.

In general, entering “0” in the case of a yes/no setting means “no”. In this example, no password is required for access to the Telnet server, although it will always ask for the callsign of the user who connects to it.

If you want to password-protect the server, change PasswordRequired to “1” and enter the password at Password=.

The CqOnly setting determines whether RTTY Skimmer Server will spot all RTTY stations it decodes, or only those determined to be CQ-ing.

The AllowAnn and AnnUserOnly settings permit you to decide if anyone can send SKIMMER/\* commands to your RTTY Skimmer Server, and if so, who. “AllowAnn=0” allows anyone who can connect to send those commands, while “1” requires you to enter a callsign in AnnUserOnly=, and only someone logged on with that callsign can command the server.

The Validation Level is a term familiar to CW Skimmer [7] and Skimmer Server[8] users. The settings are:

- 0 = Minimal
- 1 = Normal
- 2 = Aggressive
- 3 = Paranoid

The higher the value, the more rigorous the filter that determines what is accepted for posting by Telnet, and the fewer spots that are generated. Currently, the RBN recommends a value of "1".

```
[Skimmer]
Contest=DX
```

This first setting tells the program which set of contest parameters it should use when interpreting the decoded messages. These are contained in the [Contests.ini file](#) (see below). For general, non-contest use, you should set:

```
Contest=DX
```

The next section gets to the heart of RTTY SkimServer's operation, by specifying what bands it covers and which frequencies it decodes.

The settings in the next group are labeled by bandwidth. Which one the program uses depends on whether the bandwidth is 48, 96 or 192KHz. For the narrower bandwidths, there are multiple center frequencies for some bands where full coverage of RTTY frequencies is not possible with a single center frequency.

You will note that each center frequency is a little less than half the bandwidth above the bottom edge of the band. This is technically necessary because the real decodable bandwidth of each possible choice is actually a little less than the full 48, 96 or 192 KHz..

As a general proposition, you should leave these frequencies as provided in the default ini file. They are expressed in Hz.

```
CenterFreqs48=1822750,3605750,7047750,7102750,10152750,14102750,18122750,21102750,24932750,28102750,50272750,50322750
```

```
CenterFreqs96=1822750,3575000,7065000,10175500,14105000,18145500,21105000,24955500,28105000,50295500,50345500
```

```
CenterFreqs192=1822750,3575000,7065000,10175500,14105000,18145500,21105000,24955500,28105000,50295500,50345500
```

The next piece is a little tricky.

```
SegmentSel48=0101010101
SegmentSel96=011010101
```

```
SegmentSel192=011010101
```

You need to tell the program which band segments to use, out of those that are defined above. You need to count left to right in the CenterFreqs table, for the bandwidth you are using, and accept or reject segments, using 0 for reject and 1 for accept. In the example above, using 48KHz bandwidth, the receiver will listen on 80 meters, the upper 40-meter segment, 20 meters, 15 meters and 10 meters.

Using 192-KHz bandwidth, the example above would listen on 80 meters, 40 meters, 20 meters, 15 meters and 10 meters.

At most you can have a maximum of 7 segments selected at one time, depending on your receiver and your computer's capability.

The next section tells the program which frequency segments may contain RTTY signals, and instructs it to attempt to decode anything it finds there. This helps to decrease the computational load by limiting the number of CW and other digital modes that the software attempts to decode as RTTY signals. Because frequency usage varies between normal and contest situations, there are two different sets of frequencies. The line that is not in use at any given time should be disabled by placing the comment symbol, “;” in the first position.

The example below shows RTTYSegments set up for normal, non-contest operation.

```
;normal
RttySegments=1800000-1820000,3583000-3600000,7025000-7050
000,7080000-7100000,10130000-10150000,14080000-14100000,1
8100000-18110000,21080000-21100000,24910000-24930000,2808
0000-28100000,50250000-54000000
```

```
;contest
;RttySegments=1800000-1820000,3550000-3600000,7020000-710
0000,10130000-10150000,14060000-14140000,18100000-1811500
0,21060000-21150000,24910000-24930000,28060000-28150000,5
0250000-54000000
```

## **Contests.ini**

This file is also located in

*%appdata%\Afrete\Products\RttySkimServ*

Its contents define the CQ synonyms and exchange patterns that are likely to be encountered in a number of different contests.

Upon installation, Contests.ini file contains definitions for a number of RTTY contests., plus the “DX” entry that should be used outside of any contests. You can add definitions for other contests to this file using Notepad and following the instructions below. Just be sure to make a backup copy before you begin.

The profile of each contest has two portions, the first that describes the CQ equivalents and the second that describes the exchange. For example, here is the profile for the Japanese RTTY contest

```
[JARTS]
CQ1=CQ|0.4
CQ2=TEST|0.4
CQ3=QRZ|0.1
CQ4=JARTS|0.1
Exchange1=599[~]nn<~nn>|0.95
Exchange2=5NN[~]nn<~nn>|0.05
```

The title is enclosed in square brackets, and to use this profile the title must be entered without the brackets in the RttySkimServ.ini file as

```
Contest=JARTS
```

The next four lines describe how operators may be expected to solicit callers. Each of these lines must start with “CQ” and have a unique name. Next in the line is a keyword. After a “|” separator, the relative frequency of occurrence of that phrase is denoted by a decimal percentage – e.g., 0.4 means 40 percent. The total of the values of all CQ keywords should add up to 1.0.

The best way to determine these probabilities is to copy 100 stations in the contest and count the number of each format, but if you don’t have the opportunity to do that, you can make an educated guess.

The lines starting with “Exchange” describe the format of the expected **exchanges**.

The specified exchange for JARTS is to include RST and the callsign holder's age and so these are described as:

Exchange1=599[~]nn<~nn>|0.95

Exchange2=5NN[~]nn<~nn>|0.05

The exchange formats are specified using the following notation:

{a,b} = either a or b

[] = optional

<> = one or more occurrences

~ = separator (space, dash, or comma)

n = digit

@ = letter

Again every pattern must be followed by its relative frequency of occurrence, between 0 and 1, separated with '|'.

## TELNET COMMANDS

There are only a few Telnet commands that are recognized by RTTY Skimmer Server. They are:

- **SH/DX** – displays recently posted spots
- **SKIMMER/SETT** – displays the bands currently being decoded by RTTY Skimmer Server, and some other settings
- **SKIMMER/SDR** – displays the list of SDR radios whose drivers are available to the program. The driver with the asterisk (\*) is the one currently configured in RttySkimServ.ini; any others can be selected by changing the DeviceName entry above.
- **SKIMMER/RESTART** – reloads the RttySkimServ.ini and Contests.ini files and restarts decoding with the new settings. This command can have an optional argument that specifies the location of the .ini file to be loaded. For example, Skimmer/RESTART “C\another.ini” would find an ini file in the root of your C: drive when the command is sent via Telnet.

## OTHER FEATURES

### *Additional Filtering Options*

RTTY Skimmer Server comes with a copy of the Master.dta file located in

**%appdata%\Afrete\Reference\**

-- this file is used to determine the plausibility of the callsigns decoded by the program. The latest version of Master.dta is available at the Super Check Partial web site [9], and you should plan on keeping up to date.

You may discover a need to be sure you do not spot certain callsigns. For example, if high RF levels from your own transmitter cause erroneous spotting of variations on your callsign, you may want to blacklist them. To implement blacklisting, use Notepad to add the callsigns, one per line, to the text file titled **Black.lst** located in

**%appdata%\Afrete\Reference\**

You may also want to make sure that you get the quickest possible spots of certain calls, such as DXpeditions. To do this, use Notepad to add those callsigns to the **Watch.lst** file located in

**%appdata%\Afrete\UserData\**

This will give the listed calls higher priority (requiring fewer repetitions before they are spotted) for a given validation level.

## ***Additional Files***

**TelnetSrv.log** – This file, found in

**%appdata%\Afrete\Products\RttySkimServ\**

-- logs events involving the built-in Telnet server. It can be useful for debugging problems with Telnet connections.

**Spots.txt** – This file, found in the same location, is a running record of all spots made by RTTY Skimmer Server.

## **TROUBLESHOOTING**

Assuming that your receiver is operating normally with CW Skimmer Server [\[8\]](#) or another SDR program, almost all problems result from the errors in the RttySkimServ.ini file. Whenever you make any changes in the .ini file, your first step should be to make a copy of your existing .ini file and store it somewhere safe, just in case.

Double-check your settings, particularly in the Rate and CenterFreqs sections. Even a punctuation mistake or an extra digit in one of the frequencies can cause an error. Make sure you have specified a Rate, and that it is either 0, 1 or 2.

## REFERENCES

1. Streaming SIMD Extensions 3 <http://en.wikipedia.org/wiki/SSE3>
2. Quick Silver QS1R Receiver <http://www.srl-llc.com/>
3. HermesIntf <http://sourceforge.net/projects/hermesintf/files/>
4. CW Skimmer Listener <http://ol5q.nagano.cz/soft.php?page=Software>
5. SDR-Bridge <http://www.qrv.com/sdrbridge.html>
6. Reverse Beacon Network <http://www.reversebeacon.net/main.php>
7. CW Skimmer <http://dxatlas.com/CwSkimmer/>
8. CW Skimmer Server <http://dxatlas.com/SkimServer/>
9. Super Check Partial web site <http://www.supercheckpartial.com/>