

RSS User's Manual

2.0

RSS Version 3.0

JVN Communications, Inc.

<http://www.jvncomm.com>

Updated May 2005

Introduction

RSS is a software package that allows the user to create radar data files (surveillance or .srv files) for playback via a **FIRS** radar simulation tool. These files are created by specifying radar and target parameters. Radar parameters consist of geographic location, range, scan rate, and radar type. Target parameters consist of geographic location, movement, and radar return characteristics such as beacon code and correlation. Multiple targets and radars can be specified simultaneously to simulate realistic airspace and traffic situations. This same capability can allow for the simulation of extreme and special-case situations to test maximum target loads and to re-create field problems.

Radars and targets are defined by a textual language described in this manual. These commands make up a script, generally referred to as an RSS script that is interpreted by an off-the-shelf software package called Tcl. Once all commands and parameters are defined in an RSS script, the script is "executed" (run through the Tcl interpreter) producing surveillance files that can be played back through a FIRS tool as radar feeds.

RSS Scripts

The following is a suggested hierarchy of commands for RSS scripts. Although this layout is not required, it is simple, readable, and effective.

- Launch the Tcl interpreter. Must be at the top.
- Load RSS software into Tcl. Must be at the top after launching Tcl.
- Define radar locations as variables.
- Choose the projection and point of tangency.
- Define radars.
- Define static messages (RTQCs, parrots, permanent echos, status, etc.) for each radar
- Define targets.
- Execute run command. This starts the defined target activity allowing the defined radar(s) to survey the targets and create the surveillance file(s).

```

#!/bin/sh

# This next line is ignored by tcl \
exec itclsh "$0" -- ${1+"$@"} & exit 0

# Lines that start with '#' are ignored by Tcl. They are comments.

load librss.so

set ER9 +39:27:09,-74:35:031
set GB0 +39:27:09,-74:35:031

projection stereographic -tangent $ER9

radar er9 -outfile er9.srv -pos $ER9 -alt 0 -magdev 12 -type modes

# The 'source' command reads in the contents of another file
# The file er9.static_msgs contains the commands to add the static
  messages.
source er9.static_msgs

adsb gb0 -outfile gb0.ast -pos $GB0 -gbt_type 1090 -sic 1 -sac 22 \
        -status_interval 3000 -spec 1

addcmd 00:00:00 {gb0 gbttiming 1}
addcmd 00:00:00 {gb0 gbtrx 1}
addcmd 00:00:00 {gb0 gbttx 1}
addcmd 00:00:00 {gb0 gbtalarm 1}
addcmd 00:00:00 {gb0 gbtstate 1}
addcmd 00:00:00 {gb0 gbtalert 1}

# The file targets contain the target commands
source targets

run

```

Projection

A projection must be defined in every RSS script. The projection defines how planar positions are converted to GRS (Geocentric Reference System) coordinates with respect to the earth. Currently supported projection types are: stereographic, gnomonic, flat, slant and orthographic. An optional magnetic deviation may also be supplied, specifying the angle between the y-axis of the plane and true North.

Radars

Each radar must have a unique name and output file specified. If only one radar is to be simulated, the position need not be defined if latitude/longitude positions are not used to define any targets. The format for defining radars is as follows :

radar <name> -outfile <file name> [options...]

All of the options for radar definition are listed in below.

-type	Type of radar to simulate. srr, modes, asr9, srr-modes = ASR9 with ModeS (default) ibi,srr-ibi = ASR9 with IBI rse,navcan = Canadian RSE. asr11 = Digital ASR11 Radar. asr11-dod = Department of Defense ASR11. lrr = ARSR (3,4), CD2, FPS-20 cd2d = CD2D
-pos	Radar geographical position in latitude and longitude. (000:00:00,000:00:00l)
-outfile	Radar surveillance data created. End product of RSS. Usually ends in ".srv" (required)
-infile	Radar surveillance data file to "mix in" with simulated data. Could be recorded live background traffic.
-format	Output file data format. sraps,ntds = ARTS IIIA SRAP input / SCIP output format. ast,asx = Asterix format. rse = Canadian RSE format.
-az	Starting azimuth for radar scan. (0.0)
-alt	Radar altitude or elevation in feet. (0)
-PSR_rng	Primary target maximum range for RSE Canadian radar.
-SSR_rng	Secondary target maximum range for RSE Canadian radar.
-rng	Radar maximum target range.
-scan	Scan time or period of radar scan.
-magdev	Magnetic deviation for the radar.
-rngdev	Range deviation for the radar.
-azdev	Azimuth deviation for the radar.
-timedev	Time deviation for the radar.
-blipsan	Blipsan percentage.
-chans	Number of data channels for the radar.

ADSB Sensors

Option	Description (Units)	Values (Default)
-pos	Geographic position of sensor. (degrees, nautical miles)	LL or PP or XY (0,0)
-alt	Height above mean sea level of sensor. (floating point feet)	+/-20,000.0 (0.0)
-outfile	Asterix output file to create. (ASCII text file path)	(none)
-infile	Asterix input file to "mix-in" with simulated data. (ASCII text file path)	(none)
-gbt_type	Type of ADSB sensor. (ASCII text)	1090,uat,<none> <none>
-sic	Type of ADSB sensor. (ASCII text)	0-255 (0)
-sac	System Area Code (world region identifier)	0-255 (0)
-detection	System Area Code (world region identifier)	0.0-1.0 (1.0)
-drop	Percentage of messages dropped per scan. (none)	0.0-1.0 (1.0)
-status_interval	Time interval for status messages. (integer hundreds of seconds)	10-360000

DD=integer degrees

dd=floating point degrees

MM=integer minutes (0-59)

SS=integer seconds (0-59)

ss=floating point seconds

rr=floating point distance in nautical miles

LL=<latitude>,<longitude>l : (+/-)DD:MM:ss,(+/-)(D)DD:MM:ssl or (+/-)dd,(+/-)ddl

PP=rr,ddp

XY=(+/-)rr,(+/-)rr

ADSB Sensor Comands

Command	Description	Value (Default)
gbttiming	Sets timing bit value of GBT status report	0 or 1 (0)
gbtrx	Sets RX bit value of GBT status report	0 or 1 (0)
gbttx	Sets TX bit value of GBT status report	0 or 1 (0)
gbtalarm	Sets alarm bit value of GBT status report	0 or 1 (0)
gbtstate	Sets state bit value of GBT status report	0 or 1 (0)
gbtalert	Sets alert bit value of GBT status report	0 or 1 (0)

Targets

Each target must have a unique name and position. The path of the target is made up of one or more segments called "legs". Legs can be used to change target parameters at user specified time intervals. Legs are either linear (straight line) or arc (curves).

All target positions are specified relative to a two dimensional plane, defined by the "projection" command. Altitudes are processed independently. A position is specified in one on three ways: X,Y, rho,theta or lat/lon. X,Y is the default.

If the position string is suffixed with 'p', it is interpreted as rho,theta.

If the position string is suffixed with 'l', it is interpreted as lat/lon.

The format for defining targets is as follows :

target <name> {linear/arc -pos <position> [options...]} {linear/arc} [options...] ...

Options for target definition.

- bcn** Beacon code (Mode 3A) for target. A specified code will generate radar reinforced beacon messages for this target. If not specified, a search target message (or uncorrelated/correlated pair for ASR9) will be generated.
- acc** Acceleration. Linear legs only. Knots per second.
- turnrate** Turn rate. Arch legs only. Degrees per second. +=right
- hdg** Target heading in degrees. (0.0)
- vel** Target velocity in nautical miles per hour. (0.0)
- t** Start time of leg given in HH:MM:SS (00:00:00)
- dt** Delta start time of leg since beginning of previous leg given in HH:MM:SS
- et** Target end time. Target will no longer generate radar returns after this time. Given in HH:MM:SS
- pos** Initial target position. Required and valid on initial leg only. Allowed Formats :
 - Polar coordinates : R, Tp R=Range T=degrees
 - Latitude/Longitude : DD:MM:SS,DD:MM:SSl or DD.DD,DD.DDl
 - X-Y: X,Y nautical miles
- alt** Target altitude. Used in successive legs will cause "jumps" in height.
- climb** Climb rate in feet per second. Cancelled by the presence of a successive leg.
- ident**
- test** Set the test bit in a primary search target.
- rr** Toggle the radar reinforced bit in a secondary beacon message. 0,1 (1)
- val3a** Set the Mode 3A validity bit in a secondary beacon message. 0,1 (1)
- valc** Set the Mode C validity bit in a secondary beacon message. 0,1 (1)
- qual** Set the quality bits in a primary search message *. 0-3 (3)
- conf** Set the confidence bits in a primary search message *. 0-7 (5)
- elig** Set the eligibility bits in a primary search message *. 0-3(2)
- corr** Set the correlated bit in a primary search message. 0-2 (2=generate both for ASR9)
- dest** Destination position. See **-pos** for formats.
- destalt** Destination altitude in feet. Requires **-dest** argument. This altitude will be reached when the target reaches destination position.

*Consult ASR9 ICD for ARTS3A quality based on these values. (Default Values)

ADSB specific target options

Option	Description (Units)	Values (Default)
-acid	Aircraft identification. (7 character ASCII text)	(<none>)
-icao	ICAO target address identifier. (decimal integer)	0-16777215 (0)
-emitter_cat	Target's category code. (integer)	0-63 (0)
-uat_delay	Time between UAT reports. (integer hundreds of seconds)	1-10000(100)
-r1090i_delay	Time between 1090 information reports. (integer hundreds of seconds)	1-10000(500)
-r1090p_delay	Time between 1090 position reports. (integer hundreds of seconds)	1-10000(50)
-r1090v_delay	Time between 1090 velocity reports. (integer hundreds of seconds)	1-10000(50)
-adsb_type	Type of ADSB transponder. (ASCII text)	1090,uat,<none> (<none>)

Category Codes

Value	Category
0	No aircraft type information
1	Light (< 15,500 lbs)
2	Small (15,500 to 75,000 lbs)
3	Large (75,000 to 300,000 lbs)
4	High Vortex Large
5	Heavy (> 300,000 lbs)
6	High performance (> 5G Acceleration)
7	Rotocraft
8	(Unassigned)
9	Glider/Sailplane
10	Lighter than air
11	Parachutist/sky diver
12	Ultra light/hang glider/paraglider
13	(Unassigned)
14	Unmanned aerial vehicle
15	Space/transatmospheric vehicle
16	(Unassigned)
17	Surface vehicle - emergency vehicle
18	Surface vehicle - service vehicle
19	Fixed ground or tethered obstruction
20	Cluster obstacle
21	Line obstacle

Target Leg Options

Option	Description (Units)	Values (Default)
-bcn	Mode 3A beacon code (4 digit octal code)	none,00000-07777 (none)
-hdg	Heading (floating point degrees)	0.0-360.0 (0.0)
-vel	Speed (floating point nautical miles per hour)	0.0-1000.0 (0.0)
-t	Absolute start time of leg (HH:MM:SS or floating point seconds)	0.00-?? (0.0)
-dt	Offset start time from previous leg (HH:MM:SS or floating point seconds)	0.00-?? (0.0)
-et	Absolute end time of leg (HH:MM:SS or floating point seconds)	0.00-?? (never)
-pos	Starting position of target	(0,0)
-destalt	Destination altitude	
-alt	Starting altitude	
-climbrate	Climb rate	
-climb	Climb rate	
-ident	Ident bit	
-adsbident	Set the ADSB ident bit in ADSB messages	0-1 (0)
-status_code	Target status code	0-15 (0)
-test	Set the test bit on beacon targets	0-1 (0)
-rr	Set the radar reinforced bit for beacon messages	0-1 (1)
-val3a	Mode 3A validity bit	0-1 (1)
-valc	Mode C validity bit	0-1 (1)
-qual	Quality	0-7(3)
-conf	Confidence	0-5(5)
-elig	Track eligibility	0-3(2)
-rl	Run length	0-127(8)
-tis	Time in storage	0-15(1)
-corr	Send correlated primary messages for ASR sensors	0-2 (2) 2=both for ASR
-acc	Acceleration rate	
-turnrate	Turn rate	

Status Codes

Value	Status Code
0	No emergency / not reported
1	General emergency
2	Lifeguard / medical emergency
3	Minimum fuel
4	No communications
5	Unlawful interference (hijacking)
6	Downed aircraft
7-15	Reserved for future definition

Static Messages

Static messages are radar messages that a radar "sees" or transmits to receiving facilities once every scan. These are usually beacon RTQCs, search RTQCs, parrots, permanent echos, and status messages. These messages usually do not change throughout the simulation. It is possible however to modify these messages at specific times if needed. Static messages are defined hexadecimally as the words described by the ICD for a particular radar. This makes static messages difficult to construct since both the hexadecimal numerical system and bit-level radar message familiarity is required. Initial setup of static messages may require assistance from JVN Communications, Inc. Static messages are added to specific radars (after the radar is defined) in the following manner :

<radar name> add msg {<word1> <word2> ... <wordn>} <scan position>

The words are 12 bit hexadecimal values representing the words specified in the appropriate radar ICD. The words are in reverse bit order; the MSB in the ICD is the LSB in the RSS radar definition word. The scan position is a value between 0 and 1 representing where in each scan the message should be inserted. RSS will automatically insert messages with range fields in the correct position if the scan position is omitted.

Changing static messages during the simulation is done with the *chgmsg* command. This command is most effectively used in combination with the *addcmd* command that executes a command (*chgmsg*) at a user specified time. See Appendix A for examples.

Static messages can be removed during the simulation by using the *delmsg* command. This command is most effectively used in combination with the *addcmd* command that executes a command (*delmsg*) at a user specified time. See Appendix A for examples.

Running Scripts

Surveillance files are generated by "running" an RSS script. The script must be executable to do this. The *chmod* command is used to make the script executable :

chmod 755 <rss_script>. Typing the name of the script should generate a running clock output followed by the word, "done." The surveillance files specified in the script should now be present and can be played back with a FIRS tool.

Terminology

FIRS - FAA Interfacility and Radar Simulator. Also known as SIRS, TIFRS, RIFRS, and PIFRS.

Parrot – Fixed beacon transponder.

Permanent Echo – Fixed target that generates a return to a primary radar.

RSS – Radar Simulation System.

RTQC (Real Time Quality Control) – A fixed message that is generated once or twice a scan per radar.

Appendix A Examples

Normal ASR9 Radar

```
set ER9 +39:27:09,-74:35:311

radar er9 -pos $ER9 -outfile er9.srv -alt 16 -magdev 10 -type modes
```

Normal ARSR4 Radar

```
set QIE +39:49:29.04,-74:57:15.01

radar qie -pos $QIE -alt 0 -magdev 0 -outfile qie.srv -type lrr
```

Normal ADSB Sensor

```
set QIE +39:49:29.04,-74:57:15.01

adsb gb0 -outfile gb0.ast -pos $GB0 -gbt_type 1090 -sic 1 -sac 22 \
        -status_interval 3000 -spec 1
```

Normal stereographic projection

```
projection stereographic -tangent $ER9 -magdev -10
```

Stationary target

```
target TGT0001 \
    {linear -hdg 180 -vel 0 -pos 50.5,200.4p}
```

Target changing beacon code

```
target TGT0002 \
    {linear -bcn 3445 -hdg 90 -vel 220 -pos 45,90p} \
    {linear -dt 00:01:00 -bcn 7777}
```

Target flying route

```
# These may be put into a separate file and sourced
set AAA +00:00:00,00:50:001
set BBB +00:00:00,00:40:001
set CCC +00:00:00,00:30:001
set DDD +00:00:00,00:20:001
set APT +00:00:00,00:00:001

# Note that quotes are used instead of curly brackets
```

```
target ROUTEDEMO \  
  "linear -bcn 0001 -pos $AAA -alt 6000 -vel 220 -dest $BBB" \  
  "linear -dest $CCC -destalt 5000" \  
  "linear -dest $DDD -destalt 3000" \  
  "linear -dest $APT -destalt 20" \  
  "end"
```

Target circling radar

```
# Use turnrate=vel / (20 * PI * radius)  
  
# Circle radar at 200 knots range 50 nmi  
target buzz \  
  {arc -bcn 0707 -pos 50,0p -hdg 90 -vel 200 -turnrate 0.064}
```

ASR9 Search RTQC static message

```
# SRTQC at 55.06 nmi 13 ACPs  
# Uncorrelated  
er9 addmsg {0x920 0xdc4 0x00d 0xedc}  
# Correlated  
er9 addmsg {0x920 0xdc4 0x00d 0xedd}
```

ASR9 Beacon RTQC static message

```
# BRTQC at 59.06 nmi 2105 ACPs -1000 ft bc 7770  
er9 addmsg {0xec0 0xec4 0x839 0xf2a 0xff8 0x000 0xc18}
```

ASR9 Permanent echo static message

```
# PE at 1.66 nmi 1217 ACPs  
er9 addmsg {0x1b0 0x06a 0x3f7 0xddc} #uncorrelated  
er9 addmsg {0x1b0 0x06a 0x3f7 0xdd} #correlated
```

ASR9 Parrot static message

```
# Parrot at 51.98 nmi 1609 ACPs -900 ft bcn 0202  
er9 addmsg {0x6c0 0xcff 0x649 0xf6f 0x082 0x000 0xff7}
```

ASR9 Status static message

```
# Status. The second argument to addmsg is the scan position  
er9 addmsg {0x0c0 0x842 0xb00 0x800} 0.5
```

Changing status message

```
# Start with a default value
set stat_msg [er9 addmsg {0x0c0 0x842 0xb00 0x800} 0.5]

#At 10 minutes into the simulation, set the last bit of the status
message
addcmd 00:10:00 {er9 chgmsg $stat_msg "0x0c0 0x842 0xb00 0x801"}
```

Disappearing beacon RTQC

```
# Start with BRTQC on
set brtqc [er9 addmsg {0xec0 0xec4 0x839 0xf2a 0xff8 0x000 0xc18}]

# At 20 minutes, delete the message
addcmd 00:20:00 {er9 delmsg $brtqc}

# At 25 minutes, add the BRTQC back in
addcmd 00:25:00 {er9 addmsg "0xec0 0xec4 0x839 0xf2a 0xff8 0x000 0xc18"}
```