



Laboratory Utilities

User Reference Guide

October 22, 2026

JVN Tool Suite v15.2.3 Volume 7

Copyright © 2026

JVN Communications Inc.
2111 New Road, Suite 104
Northfield, NJ 08225
609-569-9255
www.jvncomm.com

TABLE OF CONTENTS

1. Overview	5
2. Getting Started	5
3. System Configuration	6
3.1. Environment Variables	6
4. Lab Utilities	7
4.1. Configuration File Utilities	9
4.1.1. SDRR Configuration Utility – exportsdrrcfg	9
4.1.2. STARS Configuration – exportsdrrcfg	13
4.1.3. Exportcomm	24
4.1.4. LabGen	24
4.1.5. EIB Configuration – eibGenGui	29
4.1.6. SDRR Connector Configuration	42
4.2. STARS CDR Utility	45
4.2.1. GUI Based Utility – STARSCDR	45
4.2.2. Command Line Utility – starscdr	47
4.3. Surveillance Data Utilities	50
4.3.1. SRV/DASR to Tracks	50
4.3.2. Track Flagging Utilities	52
4.3.3. Change SRV Utilities	53
4.3.4. Strip Utilities	54
4.3.5. Find Utilities	55
4.3.6. Loop SRV	58
4.3.7. Merge Utilities	58
4.3.8. CD2 Conversion	59
4.3.9. Scan Time	60
4.3.10. Time Slice	61
4.4. Analysis Utilities	68
4.4.1. ADS-B Network Statistics Utility	68
4.4.2. Network Capture Conversion	68
4.4.3. Monitor	69
4.4.4. Message List	70
4.4.5. Filter JVN Log	72
4.5. Scenario/Recording Utilities	73
4.5.1. ATCoach Conversion	73
4.5.2. SDRR Player	75
4.5.3. TARP	82
4.5.4. Scenario Export Utility – exportsdrr	90
4.5.5. Special Message Injection Utility	92
4.5.6. Radar Verification Tool (RVT)	93

Appendix A. Acronyms	95
-----------------------------------	-----------

TABLE OF FIGURES

Figure 1. Processor Login.....	5
Figure 2. Processor Desktop	6
Figure 3. Lab Utilities	7
Figure 4. JVN Software / Other Utilities	8
Figure 5. LabGen – DataComm	24
Figure 6. LabGen – SDRR	26
Figure 7. LabGen – Terminal	27
Figure 8. LabGen – Miscellaneous	28
Figure 9. Main eibGenGui Window	30
Figure 10. Add Harness	31
Figure 11. Individual Harness Tab	32
Figure 12. External Facilities and Interfaces	36
Figure 13. Miscellaneous Tab	40
Figure 14. Configure ZTV Displays	42
Figure 15. SDRR Connector	44
Figure 16. STARS CDR Extraction Wizard	45
Figure 17. Extraction Successful Status	47
Figure 18. Extraction Error Status	47
Figure 19: Command Line Utility starscdr	48
Figure 20: Utilities srv2tracks and dasr2tracks	50
Figure 21. Example of srv2tracks Output	51
Figure 22. ASTERIX Message Flagging Utility	52
Figure 23. Change SRV Utilities	53
Figure 24. Command Line Options for Find Utilities	56
Figure 25. Find Static Messages GUI	57
Figure 26. Loop SRV Options	58
Figure 27. Merge JVN Files	58
Figure 28. Merge SRV Files	59
Figure 29. CD2 Conversion GUI	60
Figure 30. Scan Time Options	60
Figure 31. Time Slice Utility	61
Figure 32. Time Slice GUI	62

Figure 33. Selecting a Media Device	63
Figure 34. Start Time and End Time Auto-populated	64
Figure 35. End Time Manually Changed	65
Figure 36. Execution Status Window	66
Figure 37. New Sliced Scenario	67
Figure 38. ADS-B Network Statistics	68
Figure 39. Network Capture Conversion	69
Figure 40. Monitor Utility	69
Figure 41. Message List Options	70
Figure 42. ATCoach Conversion GUI	74
Figure 43. SDRR Player Icon	75
Figure 44. SDRR Player GUIAbout:	76
Figure 45. Add Item Window	77
Figure 46. Playlist Options	79
Figure 47. Loop Options	80
Figure 48. Saving the Playlist	80
Figure 49. Save Current Playlist Dialog	81
Figure 50. TARP Icon	82
Figure 51. TARP Utility	82
Figure 52. TARP Option	83
Figure 53. Remote Host.....	85
Figure 54. Remote Password	85
Figure 55. Remote Host and User	86
Figure 56. SDRR is Launched	87
Figure 57. Playback Details	87
Figure 58. TARP Preview	88
Figure 59. AVID GUI	89
Figure 60. AVID Radar Filter.....	89

1. Overview

The JVN Tool Suite is an integrated set of software programs for simulating and displaying air traffic data with the ability to emulate and interface with a variety of air traffic control and advisory systems. The tool suite also includes an accompanying set of laboratory utilities. This includes utilities to generate SDRR configuration files for physically connected and simulated systems from ERAM, STARS, and TBFM adaptation files. Utilities are also included for data format conversion and to aid data analysis. Some of these utilities can be executed at the processor command line and some have a Graphical User Interface (GUI).

2. Getting Started

The processor with the lab utilities installed is configured to boot up to a user login screen. Users can enter a username and password, then click the login button or press **Enter** on the keyboard.

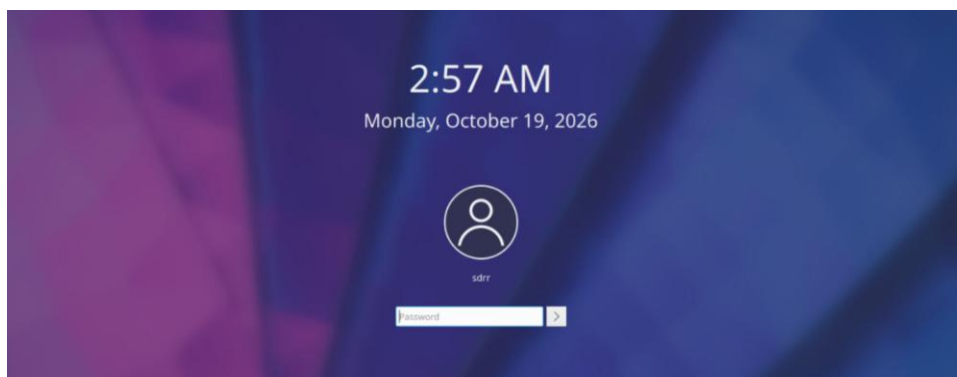


Figure 1. Processor Login

NOTE: The 'root' user does not have access to the JVN commands and utilities. If root access is needed while logged in, the user should either log out and log in again as 'root' or open a terminal window, type **su** and enter the 'root' user password.

After a short loading period the KDE desktop will appear.



Figure 2. Processor Desktop

3. System Configuration

3.1. Environment Variables

The lab utilities use several environment variables that set the locations of configuration files, scenarios, and recordings.

Table 1. Environment Variables

Variable Name	Description	Default Location
SDRR_CONFIG_PATH	Location of SDRR configuration files.	/usr/local/cfg
SDRR_SCENARIO_PATH	Location of SDRR scenario files.	/usr/local/scenarios
RECORD_PATH	Location of recording files.	/usr/local/recordings
LABUTILS_PATH	Location of labUtils	/usr/local/

4. Lab Utilities

The JVn Tool Suite includes many useful utilities for configuring simulation, investigating issues, and verifying data. These utilities are available in the software directories, `/usr/local/labUtils.X.X.X/bin` and `/usr/local/jvn.X.X.X/bin`. Example lists of the utilities are shown in the figures below. Some of the more frequently used utilities are described in this section.

```
# ls /usr/local/labUtils.15.1.0/bin/
atccomXchg          eramGen              fdstabil.eac        mergeFdkSim          socat-aix
AtcMetrics          eramGenGui           fdstabil.ead        mergeHarnesses       socat-linux
backupSqlDatabase  exportatc            fdsvcs              mergePSD.sh          starsDMSInfo
bciSessionId       exportcomm           findAdaptation       parselib              startEib.ksh
BciStmReport       exportConsoleCommands formatEramCommnd     performanceReport    startSwimServer
catwalk.ksh        exportsdrrcfg        formatOutput         PgwReport            syncSTARSAadaptation
checkApps.sh       exportsepadapt       formatPSDLog         pickListAddReport     syshealth.sh
checkArincAircraft failreport           fpConverter          relay.sh              sysinfo.pl
ClearanceTimeReport fdaddr              Front0Report         restartAPP.ksh        teardownEib.ksh
ComparePsdReport   FdcsReport          genDisplayScripts    restartSSRV.ksh       termGen
console.ksh        fdclib              generamlab           routeChecker          terminateEib.ksh.eac
copyAtcoachScenario fdknet              generateEib.ksh      rsyncAdaptationCombined
createRunwayAdapt  fdknetworkclass     genIdlsAdaptation    rvt                  timsReport
datacommATCMetrics fdksim              getAltsCounts        savePrefsets.ksh     tmsConfigure
dclAircraftId      fdksimlib           getCounts            sbomInfo              TMSConfigure
dclErrorReport     fdk ue fdsim ipl pre dss stp.ksh getNationalWx         sdrLogReport         TracerReport
DclReport          fdk ue fdsim ipl pre dss stp.ksh.healthcheck          sdrPlayerVersionSelect
displayDuAddDelay  fdk ue fdsim ipl pre fis.ksh.healthcron.sh          sdr_setup.template
displaySidsTransitions fdk ue fdsim ipl pre fis.ksh.jvnBuildHandler          sdrVersionSelect
downloadCIPF       fdk ue fdsim ipl pre sime.ksh.k3extadr          sectorAreaMap
dumpBciLog         fdk ue fdsim ipl pre sime.ksh.LabGen              selectAdaptation
dumpEadpName       fdk ue fdsim term post dss stp.ksh launchTbfm          selectFlight
dumpPayload        fdk-ue-wlboot-sime_hgi_link LogonClearanceRequestTime
ecgGen             fdnam               mergeAPA.sh          setupDisplaysGui
eibGen             fdnamlib.eac        mergeATCCOM.sh       setup_remote_displays
eibGenGui          fdnamlib.ead        showTables
```

Figure 3. Lab Utilities

```
# ls /usr/local/jvn.15.1.0/bin/
addADSBNoise        addjitter            addrsim              addStrobeSim         addTgtMsg
addANoise           addJustMsgTime       adsb2fma             adsb2ast             adsb2omr
adsb2trklist        aigClient            altysReport          apc                  assignAircraftIdentifications
assignDataCommReservedAcids
assignDataCommAndRunways
assignManualRelease
assignMode3Acodes
assignPilotPositions
assignRoutes
assignRsis
assignScript
astmsgflg
astskel
atcsrv
ATCSRV
avid
azflag
baudCheck
b2n2src
bcnflag
bcnhist
blinked
browserRecorder
browserTester
captureLogs
cass
ccuSim
cd2sbm2ast
cd2sbomr
cd2trgts
cd2srv
cd2t2gts
cfgDeviceMerge
changeFFLeadTime
chgacp
chgadsbValidity
chgalt
chgastalt
chgastime
chgbcn
chgCat34Frn6
chgchan
chgcpce
chgdsrStatus
chgfrn
chgmodcs
chgmodcsncasil
chgrrid
chgStatusBits
chgtime
cmhp2jvn
cmhpGateway
cmsInjector
createFdioScenario
createRouteClearances
createScenario
createSwimScenario
createTargetsFromTxtFile
createTrajectoryStates
createWxStationScenario
csge
csv2dasi
dasr2tracks
dcg2srv
decRaw2jvn
devidx2comm
devLoop
devtest
dis2tgt
downloadOCNUSWx
dumpair
dumpAsos
dumpdevidx
dumpIIs
dysimLogReport
ecgmulator
ecg2srv
esimImport
esimSwimImport
ExportCfg
exportCONUSWx
exportdisplaycfg
exportOCNUSWx
exportradar
exportsdrr
exportwarp
ExtractCDR
extractImage
extractsgf
favReport
filterJvnLog
findStatic
Findtime
findtime
fixpair2msgs
flightStripViewer
forcedrop
forcefpl
formatDataCommLog
formatJsn
formatPSDLog
fpalst
fpdump
freqMap
genDepLoad
genecfcfg
generateScriptDefinition
generateTailNumbers
genGroundMacros
genSurveillance
getInterface
gsst
gsst2atop
hpb
htmlviewer.sh
httpTest
ifmon2sdr
ifrecorder
IFRecordReport
ifshadow
ifstatus
iftrdt
isolatedCPUs
jvnhsi
jvnif2xml
jvnmessage
jvnmsginj
jvnph
jvnTimeslice
labconnector
log2csv
log2msg
loopsvr
mcw2msg
mergeVnFiles
mergesepadapt
mergeSRVFiles
modes
modRucTime
monitor
msglist
msgsort
msgsum
nasquestArrivalReport
nemsSession
notamLogAnalyzer
notams2Esim
notamSearchLoadTest
notamTester
nuniosrv
ofdps2sdr
parportTest
pcap2jvn
pcap2srv
pcap2svr
pcapppb
pcapstripvplan
pcRcu
pcRcuTest
pdr2csv
premoded3
premodedsctp
qagen
rawRecord
rbxm
rccnfigure
rdrcuql
recordcontroller
recordr
relaycontroller
relayGui
removeBcnOverlap
removeTargets
rmastmsg
rngfilter
routelist
rp2route
rs32srv
rtqcqual
sagen
scaleScenario
scantime
scar
scenarioListing
scenarioReport
sdr
sdrCmdInjector
sdrconnector
sdrPlayer
sdrProxy
sendMetaRfMsgs
sepadapt2cfg
serial2jvn
setflag
setmodeval
setpar
sgt2sgst
showbar
showmsg
sim2srv
simDriver
simDriverMacroInjector
simDriverTest
simPilot
sitemonitor
siteshadow
SiteShadow
siteshadowlog
smadj
smfx
snqual
snapshot
sortedscenlist
speedTest
srbox
srv2cd
srv2cenrap
srv2sim
srv2tracks
srpbb
sshfs
ssrvManager
starscdr
STARSCDR
starsmapi
stripAdsbRs
stripastbcn
stripb2
stripchan
striperr
stripField13
stripStatusBits
stripw
surfaceDriver
swac2tbm
swiftRecord
swimGateway
tarp
TARP
tbmParse
tcwClient
tcwServer
terminalServer
tfdnScenario
tgtCount
tgtIntervalReport
timeShiftTargets
timeslice
toequal
tracklist
trklist
updateInitToMacro
updateMacroSpos
updateRouteMacro
updateScriptDefinitionFile
uplevelScenarioVersion
upx2jvn
vt102
wgrb2
wxEdit
wxin
xxinInjector
xxinManager
```

Figure 4. JVN Software / Other Utilities

4.1. Configuration File Utilities

Execution of SDRR scenarios requires one or more configuration files that define the simulation environment. SDRR configuration files must correspond to the adaptation of the system(s) under test as well as the scenario and they must describe the physical and simulated interfaces required.

4.1.1. SDRR Configuration Utility – exportsdrircfg

The exportsdrircfg utility is used to generate configuration files for SDRR from ERAM, STARS, and TBFM adaptation files. The syntax to execute this utility is:

```
> exportsdrircfg adaptation --mode=MODE [options] > cfgFile.xml
```

where adaptation may refer to an ERAM adaptation directory or to an XML file (scenario.xml) where multiple ERAM and TBFM adaptation directories, STARS DMS reports, and site files may be included.

The exportsdrircfg utility accepts the following parameters:

Table 3. exportsdrircfg Utility Parameters

Parameter	Description
Valid Modes:	
--surveillance	export surveillance (radar, adsb, mlat, wam)
--nonSurveillance	export non-surveillance
--datacomm	export datacomm (avionics, dcns, tims)
--national	export national values (nadin, asdi, wmscr)
NonSurveillance Specific Options:	
--terminal --eramSim	export for dysim setup
--aig	add AIG generation to drive TFDM/TBFM (requires terminal adaptation or TBFM adaptation)
--solace	add solace TFDM msg generation (TTP, RTCS, MIS) to drive TBFM (requires TBFM adaptation)
--dysim	export for dysim setup (avionics and dysim feedback)

Parameter	Description
--noIpop	skip ipop interface (for when using a real EDDS)
--ipopPort="port"	port to connect to as edds. (for EIB mode. else will use port defined in fdnam)
--k3Port="port"	port k3 is listening on (ssrvinj)
--tdlsPort="name/port"	specify tdls port instead of default, can be used multiple times
--tcw	include the tcwInjector stanza in generated cfg
Surveillance Options:	
--terminal	export terminal surveillance only
--assc	add ASSC generation to drive TFDM (requires ground adaptation)
--sepAdaptFile="file"	override adapted sepAdapt file
--ecgpPort="port"	SDRR ecgp output port (for EIB)
--svolPort="port"	SDRR svol base output port (for EIB)
--rx	define ADSB static msgs per RX (default; agrees with static SDRR exports)
--rs	define ADSB static msgs per RS
National Specific Options:	
--eramSim	running with eramSim vs real ERAM
--tfms	add TFMS
Datacomm Specific Options:	
--noDcns	omit dcns object
--noTims	omit tims object
--virtTdls	export for virtual TDLS using simulated TIMS
--dysim=[cmu device]	export for dysim setup (noAvionics); optionally specify the cmu device to use(defaults to pipe)

Parameter	Description
--timsNotifPort="port"	for tims/pgw interface
--timsCpdlcPort="port"	for tims/pgw interface
--localFacs=adaptname	local adaptation (for afnLogonEntries)
--nsdaSite="site"	Sets the NSDA site to use for avionics. ie: KUSA
--mpg="1 or 2"	generate the mpg connection to the real dcns
Interface Related Options:	
--test	use pipes and /dev/null. (for internal testing)
--ecgEmulation	enable HGI emulation/ECGP generation.
--ecgTCP	feed ECG through TCP devices
--chanA[=facility]	(ecgTCP) specify facility to be on ERAM channel A only. default is both.
--chanB[=facility]	(ecgTCP) specify facility to be on ERAM channel B only. default is both. \${Zxx_PIPA},\${Zxx_PIPB},\${Zxx_BIPA},\${Zxx_BIPB}
--pips="pipa,pipab.."	ECG IP hostnames. used in ECG TCP mode. defaults to variable names: \${Zxx_PIPA},\${Zxx_PIPB},\${Zxx_BIPA},\${Zxx_BIPB}
--eib	use dynamic ips on interfaces where SDRR is the client (ssrvinj, ipop).
--enableDirectInterfaces	export with NAM Direct interfaces enabled
TBFM Specific Options:	
--misVersion="version"	defaults to 1.0
--rtcsVersion="version"	defaults to 2.0.0
--ttpVersion="version"	defaults to 2.0.0
Misc options:	
--userName	userName used for pipe devices

Parameter	Description
<code>--facName="facName"</code>	to limit export to given facility when supplying multiSite adaptation
<code>--verbose</code>	more options will be visible in generated config xml file

4.1.2. STARS Configuration – exportsdrrcfg

The configuration files required to connect SDRR to a STARS string must map facility adaptation to physical hardware. The exportsdrrcfg utility can generate configuration files for shadowing live data or playing back a scenario in a STARS string.

This section will describe how to use exportsdrrcfg with a STARS adaptation DMS report in combination with a device file. The device file specifies the physical connections established for the sensor interfaces between SDRR and STARS system. When the ERAM or STARS adaptation includes modifications to radar configuration, new sensors, or other data source changes, new SDRR configuration files must be generated.

To create the SDRR configuration files for STARS, a scenario.xml file listing the local ERAM adaptation and a STARS DMS report file is required.

Example scenario.xml file:

```
<scenario version="13.7.6">
  <eadp dir="./ZNY">
    <dms file="./site_adaptation.rpt_zrphl_r9d3jvn_y7_09_02_v1" name="PPP"/>
  </eadp>
  <targets>tgts.xml</targets>
  <syscmds>syscmds.xml</syscmds>
  <prefs>prefs.xml</prefs>
  <rsi>rsi.xml</rsi>
  <scriptDefinitions>scriptDefinitions.xml</scriptDefinitions>
  <airportConfigurations>airportConfigurations.xml</airportConfigurations>
  <restrictions>restrictions.xml</restrictions>
  <pilotControls>pilotControls.xml</pilotControls>
  <devctl>devctl.xml</devctl>
</scenario>
```

A scenario.xml file is automatically generated when a scenario is created in GSGT. A scenario.xml file can also be copied and modified with the absolute path to the ERAM adaptation directory and DMS report file location. Once the scenario.xml is created, a configuration file can be generated.

Use the following command to create an SDRR configuration file for injecting surveillance and non-surveillance data into STARS:

```
> exportsdrrcfg ./scenario.xml --mode=nonSurveillance --mode=surveillance --terminalMode >
phl-config.xml
```

NOTE: To generate a TCW Injector stanza in the configuration file, include the `--tcw` option. The TCW position name will be set to an asterisk (*) symbol by default. It must be changed to match the position specified in the scenario `msgs.xml` file in order to inject scripted TCW messages.

The result of the `exportsdrrcfg` utility is output to the screen unless redirected to a file using the greater than (>) symbol. In the example above, a new configuration file is created and named `phl-config.xml`.

An example of the generated configuration file is shown below. The entries in curly brackets, e.g., `${NUN2_MOD1_CHAN1}`, are defined in the device file.

```
<!-- Generated by exportsdrrcfg 13.7.6. Built Mar 25 2025 19:47:51. /usr/local/jvn.13.7.6
/bin/exportsdrrcfg ./scenario.xml mode=nonSurveillance mode=surveillance terminalMode

<eadp dir="ERAM/f50c3ny0">
  <dms file="STARS/phl/sdsphl_rheljvn_y9_02_01.rpt_p3" name="PPP"/>
</eadp>

-->
<root>
  <sources name="zny">
    <radar name="dov" device="${NUN2_MOD1_CHAN1}" type="asr11" chans="1" elev="115.00"
psrMaxRange="60" psrRangeUnits="64.00" scantime="4.90" spos="+39:06:52.00,-75:27:17.20"
ssrMaxRange="128" ssrRangeUnits="32.00">
      <brtqc acps="2104" alt="0" bcn="7777" range="59.0625"/>
      <srtqc acps="170" range="56" runlength="24"/>
      <parrot acps="3533" mode3a="1275" modec="600" range="50.0625"/>
      <permanentEcho acps="2989" modec="0" range="10.2344" runlength="24"/>
    </radar>
    <radar name="dox" device="?ttl=20&interface=${DASR_INTERFACES}" type="asr11"
elev="115.00" format="dasr" psrMaxRange="60" psrRangeUnits="64.00" sac="0x73"
scantime="4.90" sic="0x79" spos="+39:06:52.00,-75:27:17.20" ssrMaxRange="128"
ssrRangeUnits="32.00">
      <brtqc acps="2104" alt="0" bcn="7777" range="59.0625"/>
      <srtqc acps="170" range="56" runlength="24"/>
      <parrot acps="3533" mode3a="1275" modec="600" range="50.0625"/>
      <permanentEcho acps="2989" modec="0" range="10.2344" runlength="24"/>
    </radar>
```

```

<radar name="nxy" device="${NUN1_MOD5_CHAN1}" type="asr11" chans="1" elev="456.00"
psrMaxRange="60" psrRangeUnits="64.00" scantime="4.95" spos="+40:11:36.30, -75:09:12.70"
ssrMaxRange="120" ssrRangeUnits="32.00">
  <brtqc acps="3208" alt="0" bcn="1030" range="38"/>
  <srtqc acps="174" range="56" runlength="24"/>
  <parrot acps="269" mode3a="1275" modec="600" range="58.9531"/>
  <permanentEcho acps="2985" modec="0" range="1.17188" runlength="24"/>
</radar>
<radar name="ph1" device="${NUN1_MOD1}" type="asr9-modes" elev="87.00"
psrMaxRange="64" scantime="4.61" spos="+39:51:33.00, -75:16:00.30" ssrMaxRange="64">
  <brtqc acps="2102" alt="839" bcn="7770" range="59.0625"/>
  <srtqc acps="2060" range="50.0625" runlength="24"/>
  <parrot acps="781" mode3a="1273" modec="-10" range="50.0313"/>
  <parrot acps="875" mode3a="1274" modec="-10" range="44.3438"/>
  <permanentEcho acps="709" modec="0" range="2.84375" runlength="24"/>
  <streams>
    <ssr>${SIM_OUTPUT_DEVICE}?interfaces=${SIM_INTERFACES}+&{SIM_ALT_OUTPUT_
    DEVICE}?interfaces=${SIM_ALT_INTERFACES}</ssr>
    <corr>${SIM_OUTPUT_DEVICE}?interfaces=${SIM_INTERFACES}+&{SIM_ALT_OUTPUT_
    DEVICE}?interfaces=${SIM_ALT_INTERFACES}</corr>
    <uncorr>${SIM_OUTPUT_DEVICE}?interfaces=${SIM_INTERFACES}+&{SIM_ALT_OUTPUT_
    DEVICE}?interfaces=${SIM_ALT_INTERFACES}</uncorr>
    <status>${SIM_OUTPUT_DEVICE}?interfaces=${SIM_INTERFACES}+&{SIM_ALT_OUTPUT_
    DEVICE}?interfaces=${SIM_ALT_INTERFACES}</status>
    <wx>${SIM_OUTPUT_DEVICE}?interfaces=${SIM_INTERFACES}+&{SIM_ALT_OUTPUT_
    DEVICE}?interfaces=${SIM_ALT_INTERFACES}</wx>
  </streams>
</radar>
<radar name="qie" device="${NUN1_MOD2_CHAN1}?txclock=9600" type="lrr" chans="1"
elev="284.00" psrMaxRange="250" scantime="12.00" spos="+39:49:29.00, -74:57:15.00"
ssrMaxRange="250">
  <brtqc acps="2048" alt="839" bcn="7777" range="1"/>
  <srtqc acps="0" range="1" runlength="24"/>
  <parrot acps="1615" mode3a="1275" modec="730" range="64.25"/>
</radar>
<radar name="wri" device="${NUN1_MOD4_CHAN1}" type="asr11" chans="1" elev="217.00"
psrMaxRange="64" psrRangeUnits="64.00" scantime="4.80" spos="+40:02:08.50, -74:35:53.10"
ssrMaxRange="128" ssrRangeUnits="32.00">
  <brtqc acps="2104" alt="0" bcn="7777" range="59.0469"/>
  <srtqc acps="174" range="56" runlength="24"/>
  <parrot acps="1196" mode3a="1275" modec="590" range="3"/>
  <permanentEcho acps="1505" modec="0" range="11.875" runlength="24"/>
</radar>
<svol name="ads" sac="0xab" sic="0xd" svType="1">

```

```

<streams>
  <uat>${ADSB_OUTPUT_DEVICE}?interface=${ADSB_INTERFACES}</uat>
  <es1090>${ADSB_OUTPUT_DEVICE}?interface=${ADSB_INTERFACES}</es1090>
  <svol>${ADSB_OUTPUT_DEVICE}?interface=${ADSB_INTERFACES}</svol>
</streams>
<radio_station name="TTN01" lid="38" maxRange="40.00" spos="+40:16:40.11, -
74:49:10.16">
  <receiver id="0xd0260" icao="0xfaafaa" period="10.0" spos="+40:16:40.11, -
74:49:10.16" uat="0"/>
  <receiver id="0xd0261" icao="0xfaafaa" period="10.0" spos="+40:16:40.11, -
74:49:10.16" uat="0"/>
  <receiver id="0xd0262" icao="0xfaafaa" period="10.0" spos="+40:16:40.11, -
74:49:10.16" uat="0"/>
  <receiver id="0xd0263" icao="0xfaafaa" period="10.0" spos="+40:16:40.11, -
74:49:10.16" uat="0"/>
  <receiver id="0x90260" icao="0xfaafaa" period="5.0" spos="+40:16:40.11, -
74:49:10.16" uat="1"/>
</radio_station>
<radio_station name="WCHES" lid="39" maxRange="40.00" spos="+39:56:54.14, -
75:34:40.18">
  <receiver id="0xd0270" icao="0xfaafaa" period="10.0" spos="+39:56:54.14, -
75:34:40.18" uat="0"/>
  <receiver id="0xd0271" icao="0xfaafaa" period="10.0" spos="+39:56:54.14, -
75:34:40.18" uat="0"/>
  <receiver id="0xd0272" icao="0xfaafaa" period="10.0" spos="+39:56:54.14, -
75:34:40.18" uat="0"/>
  <receiver id="0xd0273" icao="0xfaafaa" period="10.0" spos="+39:56:54.14, -
75:34:40.18" uat="0"/>
  <receiver id="0x90270" icao="0xfaafaa" period="5.0" spos="+39:56:54.14, -
75:34:40.18" uat="1"/>
</radio_station>
<radio_station name="VINLD" lid="40" maxRange="40.00" spos="+39:27:24.39, -
75:06:46.82">
  <receiver id="0xd0280" icao="0xfaafaa" period="10.0" spos="+39:27:24.39, -
75:06:46.82" uat="0"/>
  <receiver id="0xd0281" icao="0xfaafaa" period="10.0" spos="+39:27:24.39, -
75:06:46.82" uat="0"/>
  <receiver id="0xd0282" icao="0xfaafaa" period="10.0" spos="+39:27:24.39, -
75:06:46.82" uat="0"/>
  <receiver id="0xd0283" icao="0xfaafaa" period="10.0" spos="+39:27:24.39, -
75:06:46.82" uat="0"/>
  <receiver id="0x90280" icao="0xfaafaa" period="5.0" spos="+39:27:24.39, -
75:06:46.82" uat="1"/>
</radio_station>

```

```

    <radio_station name="MANAH" lid="41" maxRange="40.00" spos="+39:39:58.60, -
74:18:55.08">
      <receiver id="0xd0290" icao="0xfaafaa" period="10.0" spos="+39:39:58.60, -
74:18:55.08" uat="0"/>
      <receiver id="0xd0291" icao="0xfaafaa" period="10.0" spos="+39:39:58.60, -
74:18:55.08" uat="0"/>
      <receiver id="0xd0292" icao="0xfaafaa" period="10.0" spos="+39:39:58.60, -
74:18:55.08" uat="0"/>
      <receiver id="0xd0293" icao="0xfaafaa" period="10.0" spos="+39:39:58.60, -
74:18:55.08" uat="0"/>
      <receiver id="0x90290" icao="0xfaafaa" period="5.0" spos="+39:39:58.60, -
74:18:55.08" uat="1"/>
    </radio_station>
    <radio_station name="PHL01" lid="42" maxRange="40.00" spos="+39:52:51.57, -
75:13:40.10">
      <receiver id="0xc02a0" icao="0xfaafaa" period="10.0" spos="+39:52:51.57, -
75:13:40.10" uat="0"/>
      <receiver id="0x902a0" icao="0xfaafaa" period="5.0" spos="+39:52:51.57, -
75:13:40.10" uat="1"/>
    </radio_station>
    <radio_station name="PHL02" lid="43" maxRange="40.00" spos="+39:51:39.59, -
75:15:06.48">
      <receiver id="0xc02b0" icao="0xfaafaa" period="10.0" spos="+39:51:39.59, -
75:15:06.48" uat="0"/>
      <receiver id="0x902b0" icao="0xfaafaa" period="5.0" spos="+39:51:39.59, -
75:15:06.48" uat="1"/>
    </radio_station>
    <radio_station name="ES81" lid="81" maxRange="40.00" spos="+41:01:01.92, -
75:54:50.41">
      <receiver id="0xd0510" icao="0xfaafaa" period="10.0" spos="+41:01:01.92, -
75:54:50.41" uat="0"/>
      <receiver id="0xd0511" icao="0xfaafaa" period="10.0" spos="+41:01:01.92, -
75:54:50.41" uat="0"/>
      <receiver id="0xd0512" icao="0xfaafaa" period="10.0" spos="+41:01:01.92, -
75:54:50.41" uat="0"/>
      <receiver id="0xd0513" icao="0xfaafaa" period="10.0" spos="+41:01:01.92, -
75:54:50.41" uat="0"/>
      <receiver id="0x90510" icao="0xfaafaa" period="5.0" spos="+41:01:01.92, -
75:54:50.41" uat="1"/>
    </radio_station>
    <radio_station name="A2301" lid="305" maxRange="40.00" spos="+40:24:11.30, -
74:02:38.50">
      <receiver id="0xd1310" icao="0xfaafaa" period="10.0" spos="+40:24:11.30, -
74:02:38.50" uat="0"/>

```

```

        <receiver id="0xd1311" icao="0xfaafaa" period="10.0" spos="+40:24:11.30, -
74:02:38.50" uat="0"/>
        <receiver id="0xd1312" icao="0xfaafaa" period="10.0" spos="+40:24:11.30, -
74:02:38.50" uat="0"/>
        <receiver id="0xd1313" icao="0xfaafaa" period="10.0" spos="+40:24:11.30, -
74:02:38.50" uat="0"/>
        <receiver id="0x91310" icao="0xfaafaa" period="5.0" spos="+40:24:11.30, -
74:02:38.50" uat="1"/>
    </radio_station>
</svol>
<wam name="sv02" period="3" radius="60" sac="0xbb" sic="0x02" spos="+39:51:33.00, -
75:16:00.30" svId="7">
    <streams>
        <modes>?interface=${WAM_INTERFACES}</modes>
        <es1090>?interface=${WAM_INTERFACES}</es1090>
        <svol>?interface=${WAM_INTERFACES}</svol>
    </streams>
</wam>
<asdex name="mlt" device="( ?interface=${ASDEX_MAIN_INTERFACE}+ ?interface=${ASDEX_
ALTERNATE_INTERFACE})" sac="0x00" sic="0x01" spos="+39:51:33.00, -75:16:00.30"/>
</sources>
<sources name="zny">
    <stars name="aaa" device="internal:zny-aaa" facName="aaa">
        <hostio name="zny" facName="zcn" magdev="-12.00" tangent="+40:40:25.00, -
75:27:12.00"/>
        <starsio name="hhh" magdev="-11.00" tangent="+40:11:09.00, -76:47:42.00"/>
        <starsio name="nnn" magdev="-13.00" tangent="+40:38:22.00, -73:45:59.00"/>
        <starsio name="ppp" magdev="-11.00" tangent="+39:51:33.00, -75:16:00.00"/>
        <starsio name="rrr" magdev="-12.00" tangent="+40:22:29.00, -75:58:13.00"/>
        <starsio name="www" magdev="-12.00" tangent="+41:17:41.00, -75:42:12.00"/>
    </stars>
    <stars name="bbb" device="internal:zny-bbb" facName="bbb">
        <hostio name="zny" facName="zcn" magdev="-12.00" tangent="+42:13:05.00, -
75:58:45.00"/>
        <starsio name="eee" magdev="-10.60" tangent="+42:10:28.00, -76:54:34.00"/>
        <starsio name="nnn" magdev="-13.00" tangent="+40:38:22.00, -73:45:59.00"/>
        <starsio name="www" magdev="-12.00" tangent="+41:17:41.00, -75:42:12.00"/>
    </stars>
    <stars name="eee" device="internal:zny-eee" facName="eee">
        <hostio name="zny" facName="zcn" magdev="-10.60" tangent="+42:10:28.00, -
76:54:34.00"/>
        <starsio name="bbb" magdev="-12.00" tangent="+42:13:05.00, -75:58:45.00"/>
    </stars>
    <stars name="hhh" device="internal:zny-hhh" facName="hhh">

```

```

    <hostio name="zny" facName="zcn" magdev="-11.00" tangent="+40:11:09.00, -
76:47:42.00"/>
    <starsio name="aaa" magdev="-12.00" tangent="+40:40:25.00, -75:27:12.00"/>
    <starsio name="ppp" magdev="-11.00" tangent="+39:51:33.00, -75:16:00.00"/>
    <starsio name="rrr" magdev="-12.00" tangent="+40:22:29.00, -75:58:13.00"/>
    <starsio name="www" magdev="-12.00" tangent="+41:17:41.00, -75:42:12.00"/>
</stars>
<stars name="mmm" device="internal:zny-mmm" facName="mmm">
    <hostio name="zny" facName="zcn" magdev="-13.00" tangent="+40:02:09.00, -
74:35:53.00"/>
    <starsio name="nnn" magdev="-13.00" tangent="+40:38:22.00, -73:45:59.00"/>
    <starsio name="ppp" magdev="-11.00" tangent="+39:51:33.00, -75:16:00.00"/>
</stars>
<stars name="nnn" device="internal:zny-nnn" facName="nnn">
    <hostio name="zny" facName="zcn" magdev="-13.00" tangent="+40:38:22.00, -
73:45:59.00"/>
    <starsio name="aaa" magdev="-12.00" tangent="+40:40:25.00, -75:27:12.00"/>
    <starsio name="bbb" magdev="-12.00" tangent="+42:13:05.00, -75:58:45.00"/>
    <starsio name="mmm" magdev="-13.00" tangent="+40:02:09.00, -74:35:53.00"/>
    <starsio name="ppp" magdev="-11.00" tangent="+39:51:33.00, -75:16:00.00"/>
    <starsio name="www" magdev="-12.00" tangent="+41:17:41.00, -75:42:12.00"/>
</stars>
<tcwInjector fac="ppp" user="ATBTEAM">
    <position name="*">tcp://{TCW_URL}</position>
</tcwInjector>
<stars name="rrr" device="internal:zny-rrr" facName="rrr">
    <hostio name="zny" facName="zcn" magdev="-12.00" tangent="+40:22:29.00, -
75:58:13.00"/>
    <starsio name="aaa" magdev="-12.00" tangent="+40:40:25.00, -75:27:12.00"/>
    <starsio name="hhh" magdev="-11.00" tangent="+40:11:09.00, -76:47:42.00"/>
    <starsio name="ppp" magdev="-11.00" tangent="+39:51:33.00, -75:16:00.00"/>
</stars>
<stars name="www" device="internal:zny-www" facName="www">
    <hostio name="zny" facName="zcn" magdev="-12.00" tangent="+41:17:41.00, -
75:42:12.00"/>
    <starsio name="aaa" magdev="-12.00" tangent="+40:40:25.00, -75:27:12.00"/>
    <starsio name="bbb" magdev="-12.00" tangent="+42:13:05.00, -75:58:45.00"/>
    <starsio name="hhh" magdev="-11.00" tangent="+40:11:09.00, -76:47:42.00"/>
    <starsio name="nnn" magdev="-13.00" tangent="+40:38:22.00, -73:45:59.00"/>
</stars>
<eramsim name="zny" autoTR="1" quietMode="0">
    <stars name="aaa" device="internal:zny-aaa" afsi="0" f3c="1" magdev="-12.00"
naps="1" tangent="+40:40:25.00, -75:27:12.00"/>

```

```

    <stars name="bbb" device="internal:zny-bbb" afsi="0" f3c="1" magdev="-12.00"
naps="1" tangent="+42:13:05.00,-75:58:45.00"/>
    <stars name="eee" device="internal:zny-eee" afsi="0" f3c="1" magdev="-10.60"
naps="1" tangent="+42:10:28.00,-76:54:34.00"/>
    <stars name="hhh" device="internal:zny-hhh" afsi="0" f3c="1" magdev="-11.00"
naps="1" tangent="+40:11:09.00,-76:47:42.00"/>
    <stars name="mmm" device="internal:zny-mmm" afsi="0" f3c="1" magdev="-13.00"
naps="1" tangent="+40:02:09.00,-74:35:53.00"/>
    <stars name="nnn" device="internal:zny-nnn" afsi="0" f3c="1" magdev="-13.00"
naps="1" tangent="+40:38:22.00,-73:45:59.00"/>
    <stars name="ppp" device="{NUN1_MOD6}" afsi="0" f3c="1" magdev="-11.00" naps="1"
tangent="+39:51:33.00,-75:16:00.00"/>
    <stars name="rrr" device="internal:zny-rrr" afsi="0" f3c="1" magdev="-12.00"
naps="1" tangent="+40:22:29.00,-75:58:13.00"/>
    <stars name="www" device="internal:zny-www" afsi="0" f3c="1" magdev="-12.00"
naps="1" tangent="+41:17:41.00,-75:42:12.00"/>
    <externalArts>
        <entry localName="alb">zbw:ala</entry>
        <entry localName="bd1">zbw:bda</entry>
        <entry localName="pvd">zbw:nca</entry>
        <entry localName="syr">zbw:sya</entry>
        <entry localName="acy">zdc:acy</entry>
        <entry localName="dov">zdc:dov</entry>
        <entry localName="nvf">zdc:nvf</entry>
        <entry localName="uct">zdc:pct</entry>
        <entry localName="jst">zob:jjj</entry>
        <entry localName="roc">zob:rrr</entry>
    </externalArts>
    </eramsim>
    <dasi name="ppp-acy" device="{DASI_DEVICE1}:110" type="fa10454" value="29.90"/>
    <dasi name="ppp-ph1" device="{DASI_DEVICE2}:110" type="fa10454" value="29.90"/>
    <etms name="tmu" device="{ETMS_DEVICE}:19200" starsid="111"/>
</sources>
</root>

```

```

    <stars name="ppp"
device="cmhp://;siteId=PHL&primary=PPP&serverMode&interface={CMHP_MAIN_INTERFA
CE}:zny-1|cmhp://;siteId=PHL&primary=PPP&serverMode&interface={CMHP_ALT_
INTERFACE}:zny-2|cmhp:// ;siteId=PHL&primary=PPP&serverMode&interface=$
{CMHP_MAIN_INTERFACE}:zny-3|cmhp://;siteId=PHL&primary=PPP&serverMode&
interface={CMHP_ALT_INTERFACE}:zny-4" afsi="0" f3c="1" magdev="-11.00" naps="1"
tangent="+39:51:33.00,-75:16:00.00"/>

```

NOTE: *exportsdrrcfg* can parse DMS files to generate CMHP IFDT output.

An example of a device file is shown below:

```
NUN1_MOD1 sirs17s1@/dev/srr0
NUN1_MOD2 sirs17s1@/dev/lrr1
NUN1_MOD2_CHAN1 sirs17s1@/dev/lrr1:1
NUN1_MOD3 sirs17s1@/dev/lrr2
NUN1_MOD3_CHAN1 sirs17s1@/dev/lrr2:1
NUN1_MOD4 sirs17s1@/dev/srr3
NUN1_MOD4_CHAN1 sirs17s1@/dev/asr11-3:1
NUN1_MOD5 sirs17s1@/dev/srr4
NUN1_MOD5_CHAN1 sirs17s1@/dev/asr11-4:1
NUN2_MOD1 sirs17s1@/dev/srr5
NUN2_MOD1_CHAN1 sirs17s1@/dev/asr11-5:1
NUN2_MOD2 sirs17s1@/dev/lrr6
NUN2_MOD3 sirs17s1@/dev/lrr7
NUN2_MOD4 sirs17s2@/dev/srr0
NUN2_MOD5 sirs17s2@/dev/srr1
NUN2_MOD6 sirs17s2@/dev/srr2
NUN3_MOD1 sirs17s2@/dev/srr3
NUN3_MOD2 sirs17s2@/dev/srr4
NUN3_MOD3 sirs17s2@/dev/srr5
NUN3_MOD4 sirs17s2@/dev/lrr6
NUN3_MOD5 sirs17s2@/dev/lrr7
NUN3_MOD6 /dev/null

NUN1_MOD6 file:///dev/ifa0?txclock=2400&rxclock=2400

ADSB_INTERFACES enp7s4f1:1,enp7s4f1:2

WAM_INTERFACES enp8s4f0,enp8s4f1

DASR_INTERFACES enp8s4f0

ASDEX_MAIN_INTERFACE enp7s4f1:4,enp7s4f1:6
ASDEX_ALTERNATE_INTERFACE enp7s4f1:5,enp7s4f1:7

SIM_OUTPUT_DEVICE
SIM_INTERFACES enp4s0f1
SIM_ALT_OUTPUT_DEVICE
SIM_ALT_INTERFACES enp7s4f0

CMHP_MAIN_INTERFACE enp4s0f1
```

```
CMHP_ALT_INTERFACE enp7s4f0

DASI_DEVICE1 sirs17s1@dasi:/dev/ttyS2
DASI_DEVICE2 sirs17s1@dasi:/dev/ttyS3
DASI_DEVICE3 dasi:/dev/null
DASI_DEVICE4 dasi:/dev/null
DASI_DEVICE5 dasi:/dev/null
DASI_DEVICE6 dasi:/dev/null
DASI_DEVICE7 dasi:/dev/null
DASI_DEVICE8 dasi:/dev/null
DASI_DEVICE9 dasi:/dev/null
DASI_DEVICE10 dasi:/dev/null

ETMS_DEVICE etms:/dev/ttyS2

TCW1_IP X.X.X.X
TCW_URL X.X.X.X:XXXX

LIVE_QIE multi://X.X.X.X:XXXX?interface=en01
LIVE_PHL multi://X.X.X.X:XXXX?interface=en01
```

Device files are created by JVN and located in the /usr/local/etc directory of the SDRR processor. When using a configuration that was created with a DMS report file, a device file must also be used to direct the radars/sensors to the correct input on the STARS system. In this device file example, NUN1_MOD1 points to sirs17s1:/dev/srr0 and based on the configuration file the radar input for NUN1_MOD1 in this adaptation is PHL.

NOTE: *If a remote box is responsible for sending ADSB data, the device file should contain the variable “ADSB_OUTPUT_DEVICE”. If the device file does not call out a remote box, the variable in the configuration file will be ignored.*

Below is an example of the syntax to run an SDRR scenario using a configuration file created with a DMS report and a device file. In this example, the device file has a custom name and is explicitly called using the --deviceFile option.

```
> sdr /usr/local/cfg/phl-config -s /usr/local/scenarios/PHL-scenario/sdr.xml
--deviceFile=/usr/local/etc/sirs17-devices
```

4.1.3. Exportcomm

The exportcomm utility generates one set of ECG communication files based on the ERAM local adaptation and set of existing sepadapt files.

Usage:

```
exportcomm adaptation [-p pipFile] [-b bipFile]
```

4.1.4. LabGen

LabGen is a utility used to generate SDRR configuration files to pair with ERAM Test Beds.

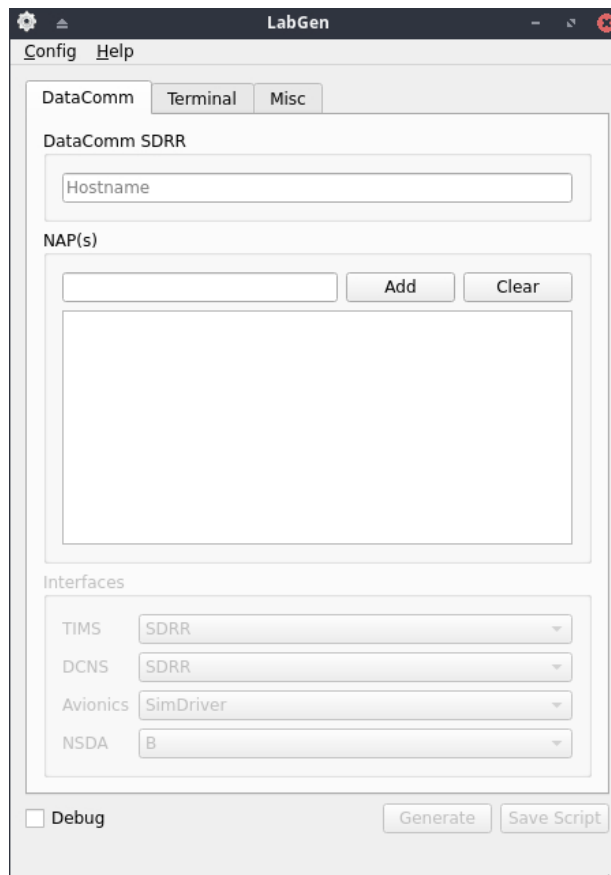


Figure 5. LabGen – DataComm

Datacomm

DataComm SDRR

This field is for the main datacomm SDRR's hostname

NAP(s)

If applicable, users can add/remove the related NAP(s) associated with their ERAM Test Bed(s).

TIMS

TIMS can be set to either SDRR simulated or LIVE.

DCNS

DCNS can be set to either SDRR simulated, MPG1, or LIVE.

Avionics

Avionics can be set to either SDRR, simDriver, or LIVE.

NSDA

NSDA can be set to either A, B, or C.

General Options

Debug

This option adds in the GDB debugger and can capture information if the program runs into any problems.

Generate

When everything else is filled out, this button is used to generate the files.

Save Script

This option allows the users to save their current configured SDRR script.

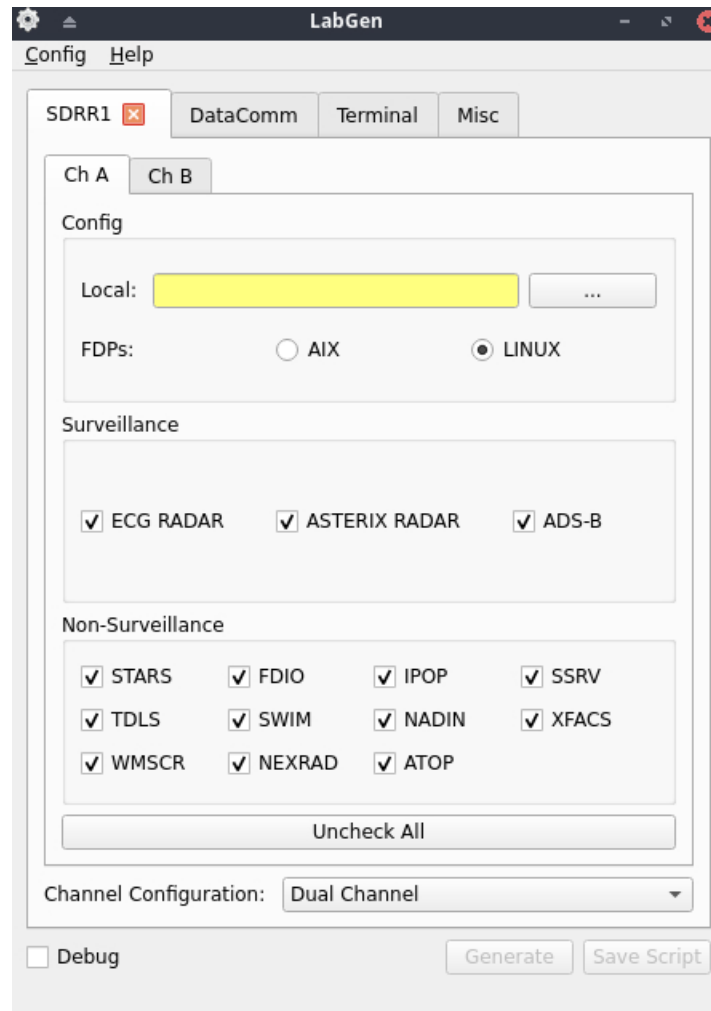


Figure 6. LabGen – SDRR

SDRR Tab(s)

Under Config, there is an option to add SDRR. Depending on the users' lab conditions, more than one SDRR may be required.

CHA / CHB

These channels either be the same or different depending on if the lab requires different sites per channel or not.

Local

This is where the ERAM local is added for the channel(s).

FDPs

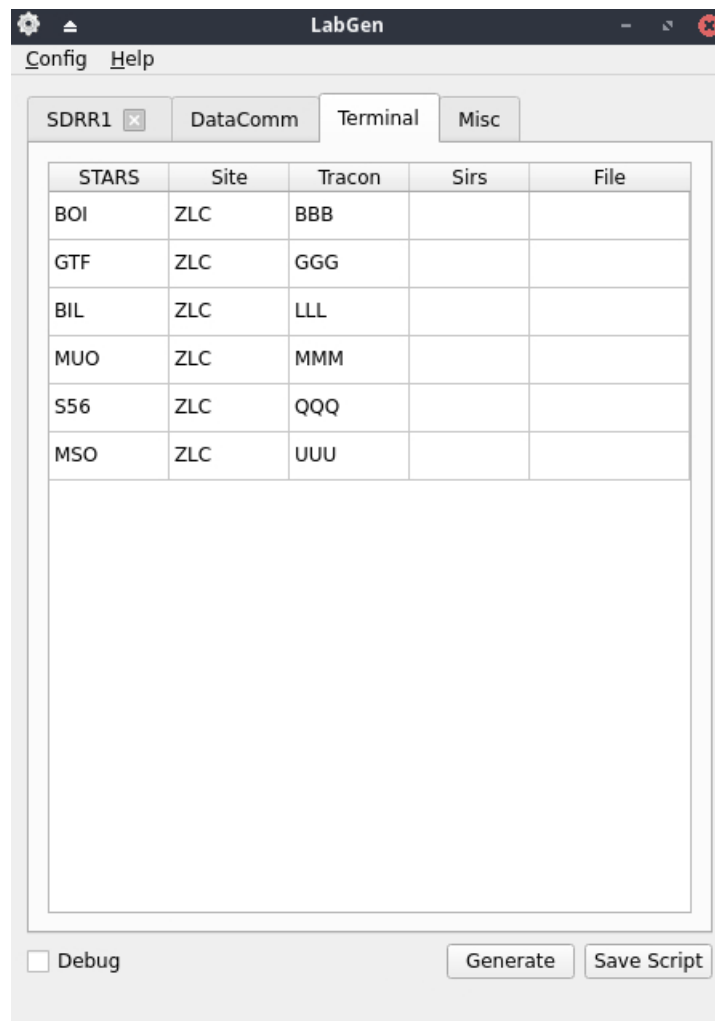
This option is used to select whether the user needs AIX or LINUX

Surveillance / NonSurveillance

These options if checked will all be SDRR simulated. Only uncheck these options if the user requires LIVE instead of simulated.

Channel Configuration

By default, dual channel configuration is selected. This means that configs will be generated for both CHA and CHB. Dual channel split configuration allows for different ERAM locals to be used on CHA vs CHB. Single channel A or B allows for only one channel's configuration to be generated.



STARS	Site	Tracon	Sirs	File
BOI	ZLC	BBB		
GTF	ZLC	GGG		
BIL	ZLC	LLL		
MUO	ZLC	MMM		
S56	ZLC	QQQ		
MSO	ZLC	UUU		

Figure 7. LabGen – Terminal

Terminal Table

This table will be auto prepopulated based on the sites selected on the SDRR(s). Then for each STAR, a Sirs and report file can be assigned.

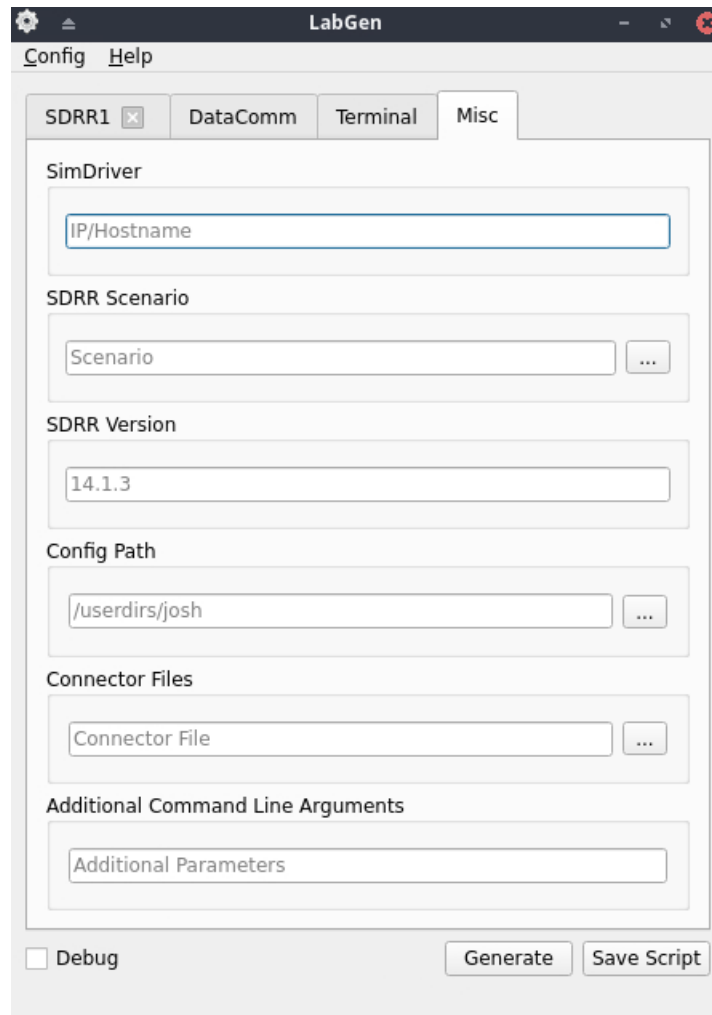


Figure 8. LabGen – Miscellaneous

Miscellaneous

SimDriver

This field is for adding the connecting simDriver's IP/Hostname.

SDRR Scenario

This field is for adding an SDRR scenario that will run without simDriver.

SDRR Version

In this field, the SDRR version can be hard-coded in.

Config Path

Enter the location to the configs if the user isn't running from the top-level

Connector Files

Enter the location to the connector files needed for Terminal.

Additional Command Line Arguments

Enter any additional needed arguments.

4.1.5. EIB Configuration – eibGenGui

The configuration files needed to connect SDRR and an ERAM-in-a-Box (EIB) must be perfectly matched for addresses of facilities, remote displays, and port numbers of all data feeds. To aid in the generation of these configuration files, the eibGenGui utility allows users to generate an entire set of SDRR and EIB configuration files in one process.

To generate the configuration files using eibGenGui, the following is entered at the processor command line:

```
> eibGenGui
```

The following GUI is displayed.

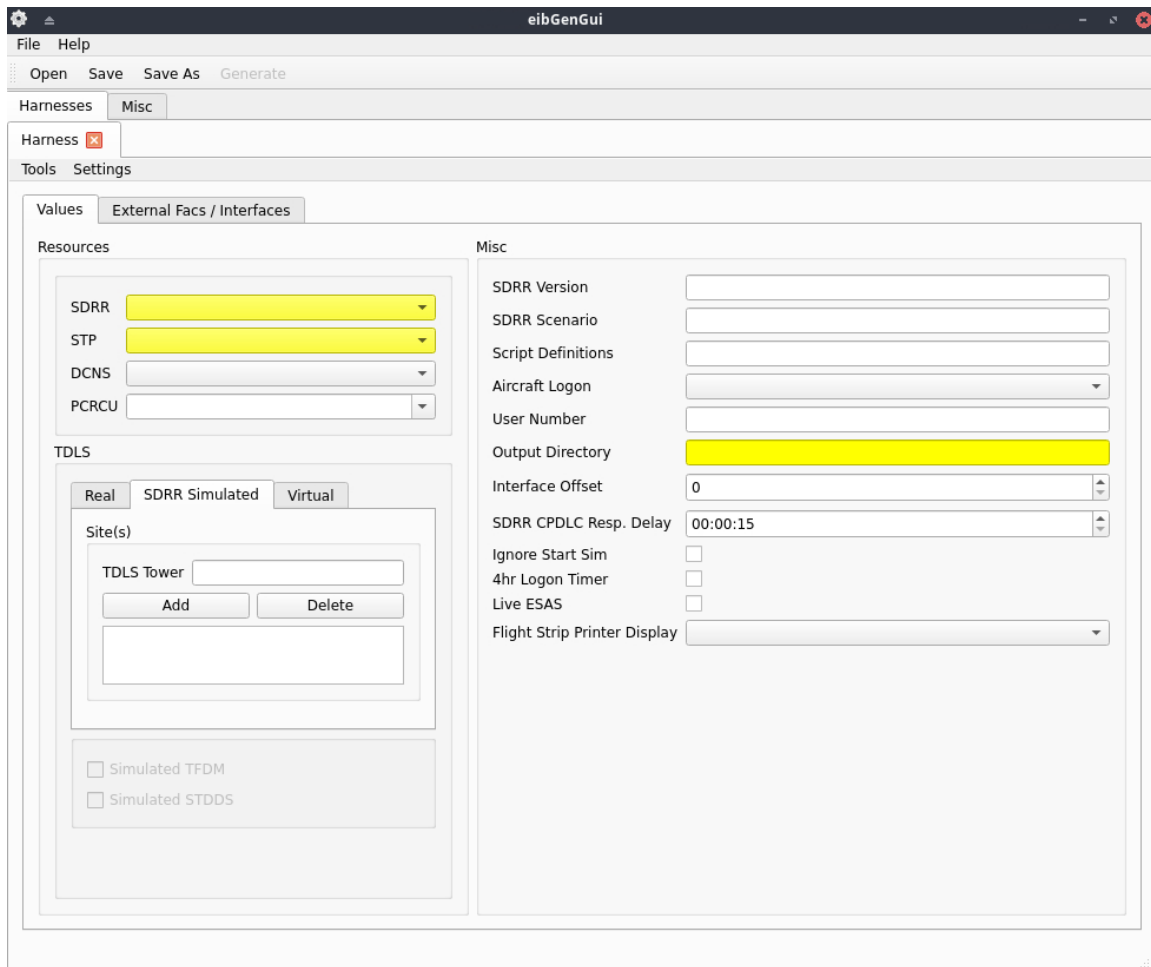


Figure 9. Main eibGenGui Window

At least one harness must be created in the “Harnesses” tab. To create a harness, select **File -> Add harness**. Now by default, one harness is already set up.

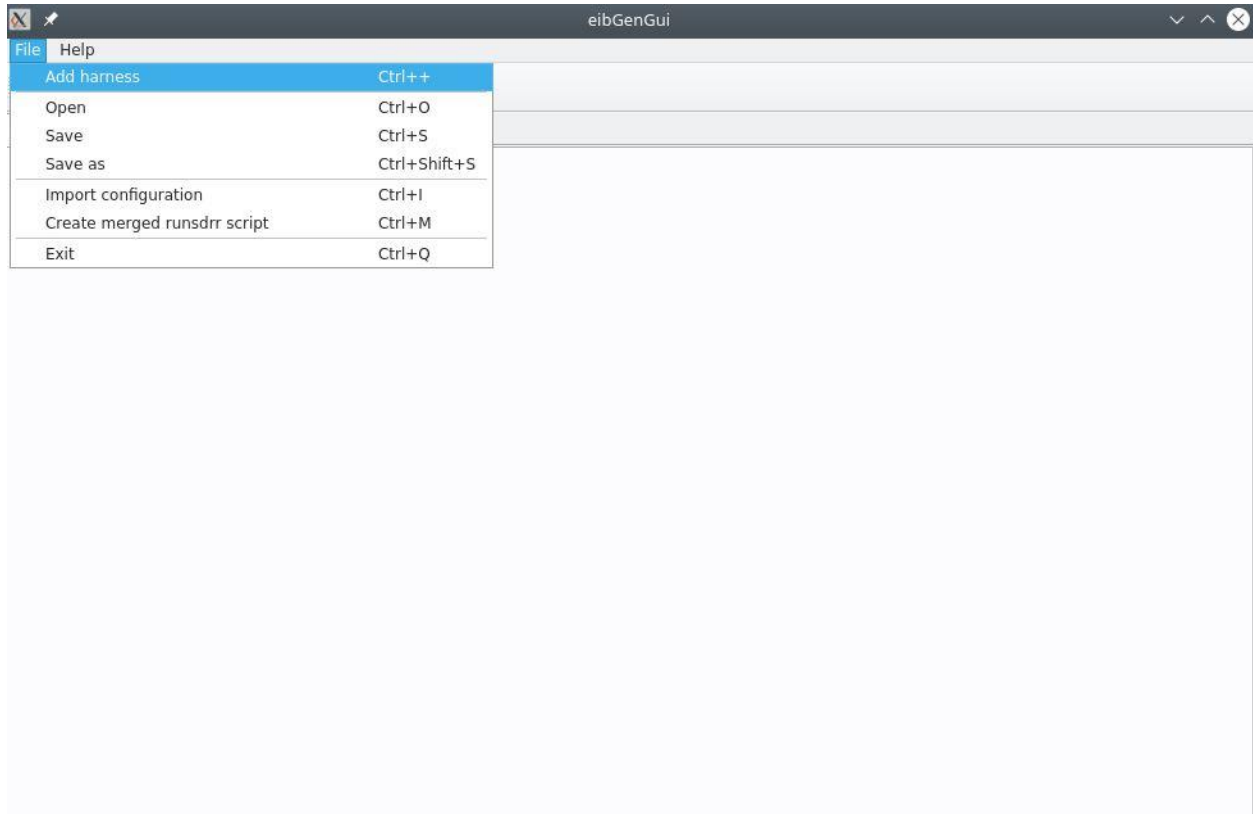
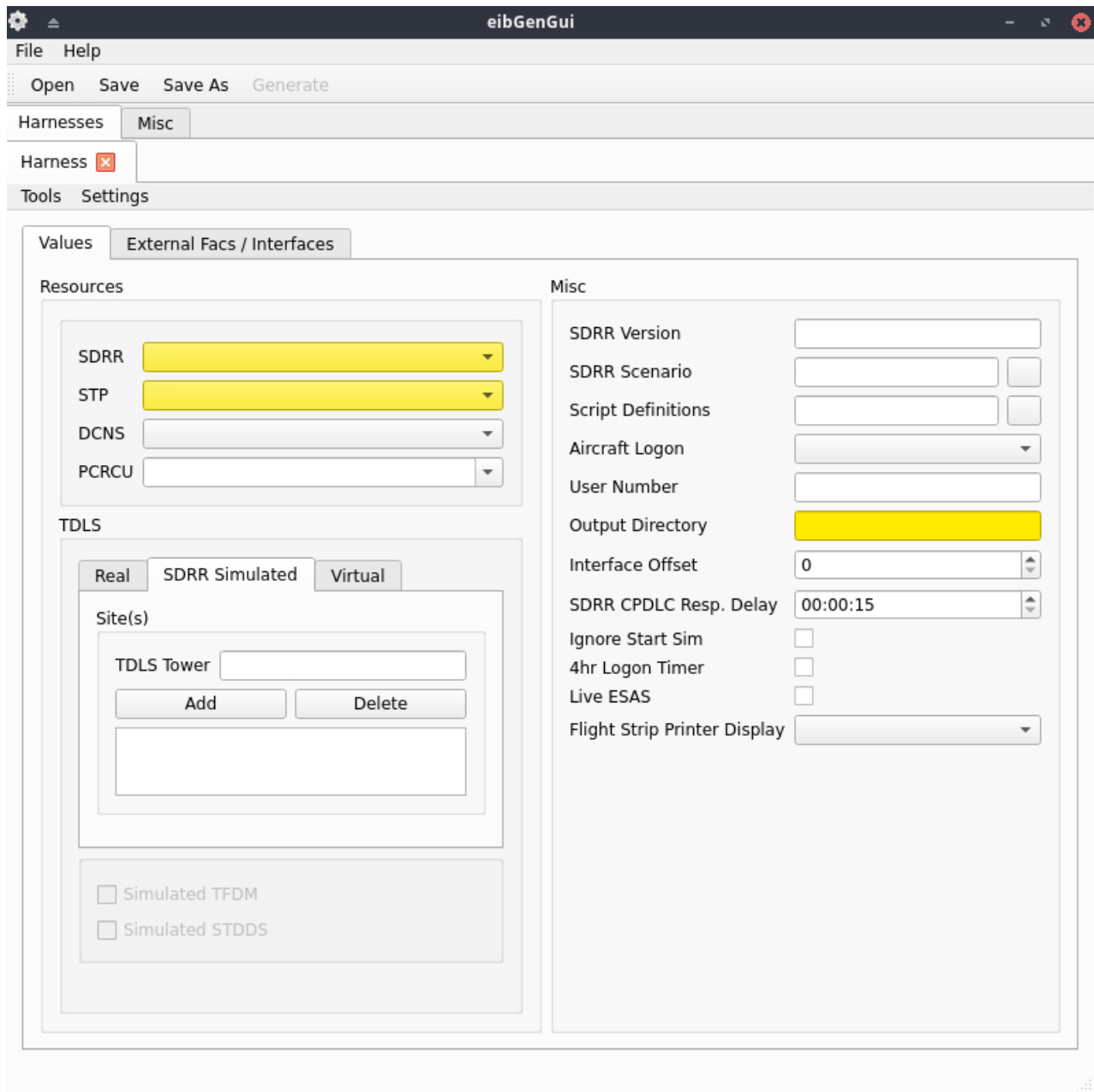


Figure 10. Add Harness

Each harness that is added will have a separate tab within the “Harnesses” tab. And within the tab for each individual harness, there will be a tab labeled “Values” and a tab labeled “External Facs / Interfaces”.



eibGenGui

File Help

Open Save Save As Generate

Harnesses Misc

Harness ✕

Tools Settings

Values External Facs / Interfaces

Resources

SDRR ▼

STP ▼

DCNS ▼

PCRCU ▼

TDLS

Real SDRR Simulated Virtual

Site(s)

TDLS Tower

Add Delete

☐ Simulated TFDM

☐ Simulated STDDS

Misc

SDRR Version

SDRR Scenario ☐

Script Definitions ☐

Aircraft Logon ▼

User Number

Output Directory ▼

Interface Offset

SDRR CPDLC Resp. Delay

Ignore Start Sim ☐

4hr Logon Timer ☐

Live ESAS ☐

Flight Strip Printer Display ▼

Figure 11. Individual Harness Tab

The “Values” tab of each individual harness includes the following options and settings:

Resources

SDRR

MANDATORY. From the drop-down menu, the desired SDRR processor may be selected.

STP

MANDATORY. From the drop-down menu, the desired EIB STP processor may be selected.

DCNS

When running with a NAP, the desired DCNS may be selected from the drop-down menu.

PCRCU

If running with LIVE FDIO, a Personal Computer-Remote Control Unit (PCRCU) may be selected from the drop-down menu.

Real TDLS

Users have the option of selecting REAL, SIMULATED, or VIRTUAL TDLS, if applicable. The desired live TDLS may be selected once the REAL TDLS box is checked.

SDRR Simulated TDLS

TDLS Tower

The tower to be simulated must be entered with the ICAO name; e.g., KBOI, KSLC, KMSP, KDEN.

Add

Once the tower has been added, the **Add** button may be used to include the simulated TDLS to the configuration and a subsequent TDLS may be entered.

Delete

Once a simulated TDLS has been added, the **Delete** button allows users to remove the simulated TDLS from the configuration.

Virtual TDLS

Virtual is similar to live but everything will be displayed virtually. These run separately from SDRR.

TDLS Tower

The tower to be simulated virtually must be entered with the ICAO name; e.g., KBOI, KSLC, KMSP, KDEN.

Add

Once the tower has been added, the **Add** button may be used to include the simulated virtual TDLS to the configuration and a subsequent TDLS may be entered.

Delete

Once a simulated virtual TDLS has been added, the **Delete** button allows users to remove the simulated virtual TDLS from the configuration.

Simulated TFDM

If applicable, users have the option of configuring a simulated TFDM when running with a live TDLS.

Simulated STDDS

If applicable, users have the option of configuring a simulated STDDS.

Misc

SDRR Version

If applicable, user can input a SDRR version other than the default on the **SDRR** box previously selected.

SDRR Scenario

If applicable, user can input an exported SDRR scenario to be loaded.

Script Definitions

If applicable, define location for Script Definitions:
Ex: /usr/local/scenarios/scenario/scriptDefinition.xml

Aircraft Logon

User can set the NSDA Logon Address to be used from the drop-down box.

User Number

Users may input a user number specifying which port offsets to use when generating. If left blank, the user number of the current logged on user will be used.

Output Directory

MANDATORY. This is where the user specifies the directory name for the configuration files.

Interface Offset

Default setting is 0. Users can select SDRR sub-interface(s): if0, if1, etc. This is to prevent conflicts with other users. Interface Offset values are 0 through 9 for most SDRR processors.

SDRR CPDLC Resp. Delay

Default setting is 15 seconds; users can change this setting to their desired value.

Ignore Start Sim

Select to have SDRR ignore the SSIM from SSRV. Requires users to press start in SDRR when ready to run.

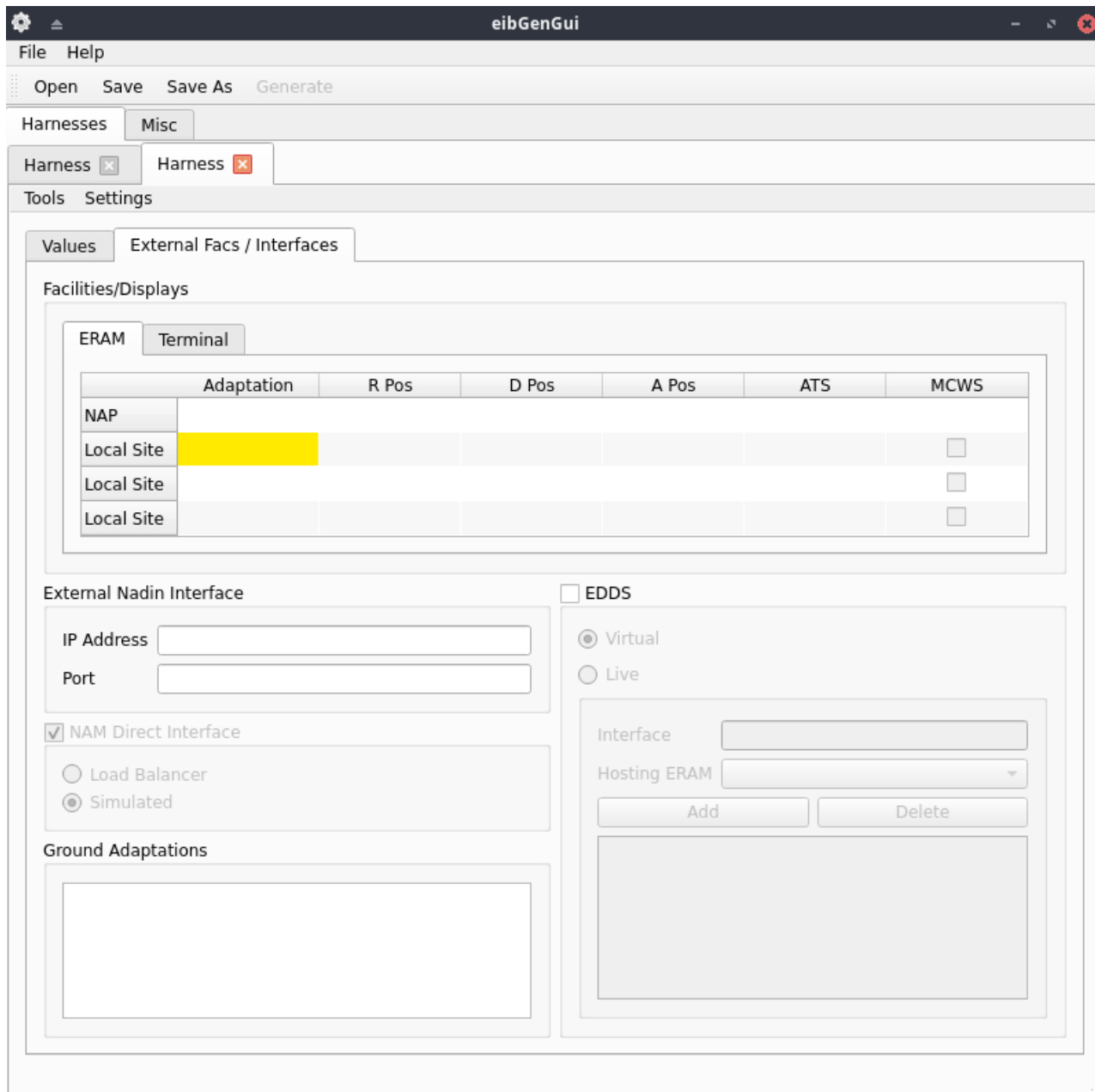
4 Hour Logon Timer

If this option is selected, it sets the <Logon_Close_Accepted_Interval> in cmapparamters.xml to a value of 4 hours. This is adaptable per local.

Live ESAS

If this option is selected, the user will have the SWIM server loaded on the STP and be able to connect to live HADDs, if applicable.

The “External Facs / Interfaces” tab of each individual harness includes the following options and settings:



	Adaptation	R Pos	D Pos	A Pos	ATS	MCWS
NAP						
Local Site						☐
Local Site						☐
Local Site						☐

Figure 12. External Facilities and Interfaces

External Nadin Interface

This is to be used if running with **LIVE NADIN**.

IP Address

IP address that EIB will use to connect to live NADIN.

Port

Port that EIB will use to connect to live NADIN.

NAM Direct Interface

This is to be used if running with the CAATS Interface.

Load Balancer

Selected if running with the FNTB Load Balancer.

Simulated

Selected if running with SDRR simulated connection.

EDDS Live Interfaces**Virtual EDDS**

This setting allows users to enter connecting SDRR and VNC start port information.

Live EDDS

This setting allows users to enter connecting to live EDDS.

Interface

IP address of the EDDS to which EIB will connect.

Hosting ERAM

This drop-down box populates after the user inputs an **Adaptation** within the **Local Release** of the **Facilities/Displays** section. The user will be able to select from any of the Local Releases added within the **Facilities/Displays** section.

Add

Once an **Interface** and **Hosting ERAM** dialog boxes have been filled out, the user then can select **Add** and it will populate within the blank dialog box.

Delete

User has the option to **Delete** an **Interface/Hosting ERAM** once added to the blank dialog box.

Ground Adaptations

User can add **Ground Adaptations** in this section. When user right clicks within this section, they have the following options.

Add

User can specify a **Ground Adaptation**.

Delete

User can delete a **Ground Adaptation**.

Terminal Facilities

Select this option to connect a live STARS string to EIB. When user right clicks within this section, they have the following options.

Add

User can specify a specific STARS adaptation.

STARS File

Users can input the path of STARS adaptation in the format of a STARS DMS report, *.rpt, file.

Lab/String

User selects a SIRS device from the drop-down box.

Hosting Eram Facility

User selects the ERAM facility from the drop-down box. This drop-down box populates after the user inputs an **Adaptation** within the **Local Release** of the **Facilities/Displays** section.

Interfacility Device

User can select either 0 or 1.

Simulated

User can set the terminal facility to simulated.

Edit

User can edit an added **Terminal Facility**.

Delete

User can delete an added **Terminal Facility**.

Facilities/Displays

Here the user can add the **Adaptation** needed to create the **Local Site** release harnesses and the **NAP** harness. Once a **Local Site** release is selected, the user can add the following options:

R Pos

Radar Position. Users can input the sectors for which they want **R Pos** displays. Users can input multiple sectors separated by commas with no spaces. Ex: 5,12,85

D Pos

Data Position. Users can input the sectors for which they want **D Pos** displays. Users can input multiple sectors separated by commas with no spaces. Ex: 5,12,85

A Pos

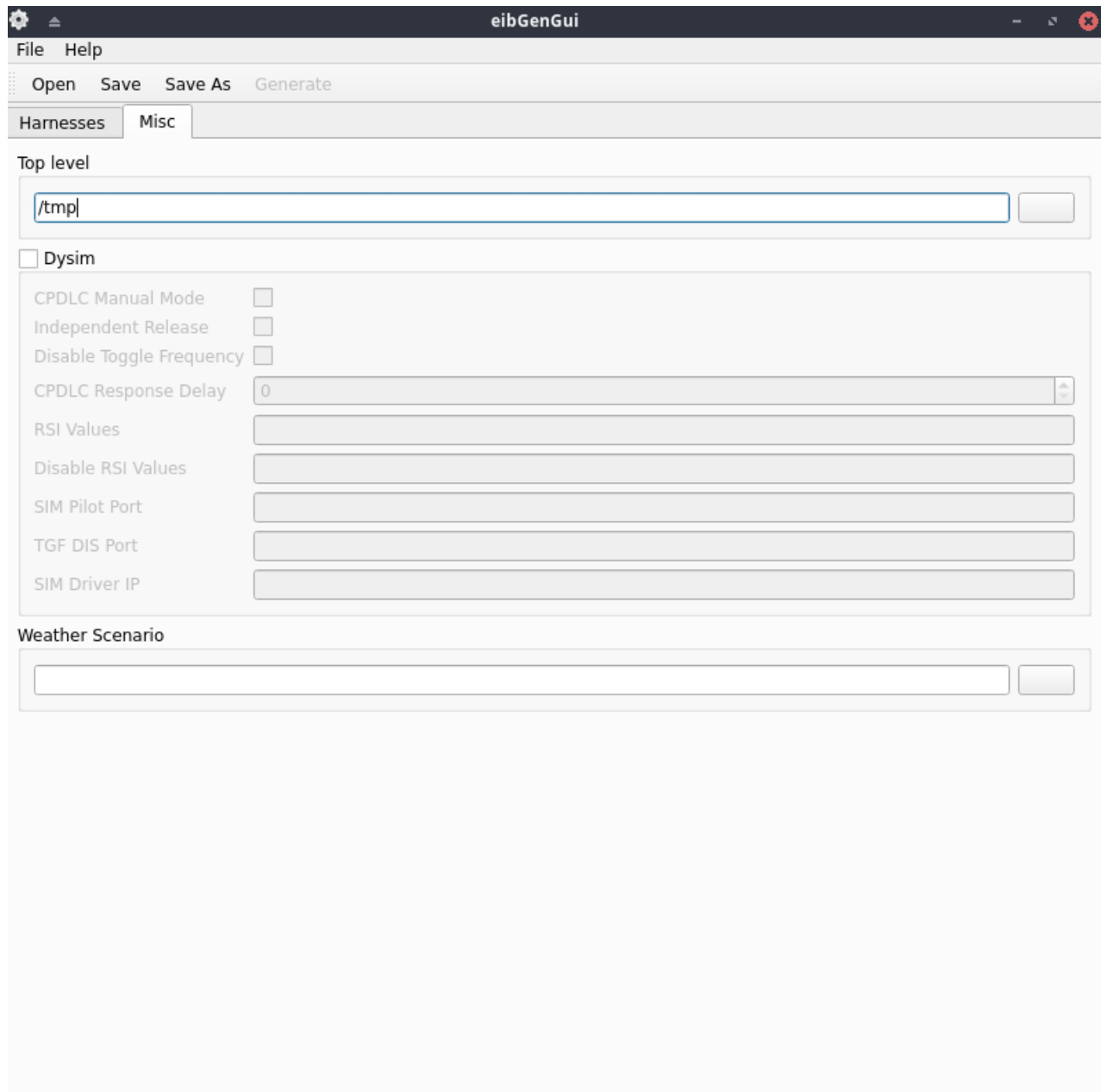
User can input the sectors for which they want **A Pos** displays. Users can input multiple sectors separated by commas with no spaces. Ex: 5,12,85

ATS

Air Traffic Specialist. Users can input the specialist position for which they want **ATS** displays. Users can input multiple specialist positions separated by commas with no space. Ex: s1,e1,w1

MCWS

Monitor and Control Workstation. Users can select whether they want an MnC. If this option is selected, it needs to be selected for every local release input. Users will need PSIM and SSIM manually.



The screenshot shows the 'eibGenGui' application window. The title bar includes a gear icon, a maximize icon, and a close icon. The menu bar contains 'File' and 'Help'. The toolbar has 'Open', 'Save', 'Save As', and 'Generate' buttons. Below the toolbar are two tabs: 'Harnesses' and 'Misc', with 'Misc' being the active tab. The 'Top level' section contains a text input field with the value '/tmp/' and a button to its right. Below this is a checkbox labeled 'Dysim'. Under the 'Dysim' section, there are several settings: 'CPDLC Manual Mode' (checkbox), 'Independent Release' (checkbox), 'Disable Toggle Frequency' (checkbox), 'CPDLC Response Delay' (text input with value '0' and a spinner), 'RSI Values' (text input), 'Disable RSI Values' (text input), 'SIM Pilot Port' (text input), 'TGF DIS Port' (text input), and 'SIM Driver IP' (text input). At the bottom, the 'Weather Scenario' section contains a text input field and a button to its right.

Figure 13. Miscellaneous Tab

The miscellaneous, “Misc”, tab includes options and settings that apply to the overall configuration, not any individual harness.

Top Level

Users need to specify their top level. Ensure your user has the necessary permissions for that directory.

Dysim

Check if running with simDriver.

CPDLC Manual Mode

These are global settings that will impact every target in the simDriver scenario. This setting allows users to set the flights to manual mode.

Independent Release

These are global settings that will impact every target in the simDriver scenario. This setting allows users to set the flights in the scenario to independent release.

Disable Toggle Frequency

These are global settings that will impact every target in the simDriver scenario. User can select to not have flights toggle on frequency automatically.

CPDLC Response Delay

These are global settings that will impact every target in the simDriver scenario. Default setting is 30 seconds; users can change this setting to their desired value.

RSI Values

This setting allows users to input the RSI of the flights that they want to be loaded from their scenario.

Disable RSI Values

Opposite from above, user can input an RSI of the flights that they want to leave out of the scenario.

SIM Pilot Port

This setting allows users to input a port value for another user to connect a simPilot to that port.

TGF DIS Port

Select 4-digit port number for DIS TGF Target Position Reports.

SIM Driver IP

User inputs where they are running their simDriver from. Users can input the IP address or hostname of that box.

Weather Scenario

This setting allows users to insert a weather scenario from anywhere on their machine.

Once all the required fields are entered into each individual harness and the miscellaneous tab, Generate becomes available. Click Generate to create the set of configuration files. Once the configuration files are generated, displays can be configured for the virtual test lab (VTL) under tools.

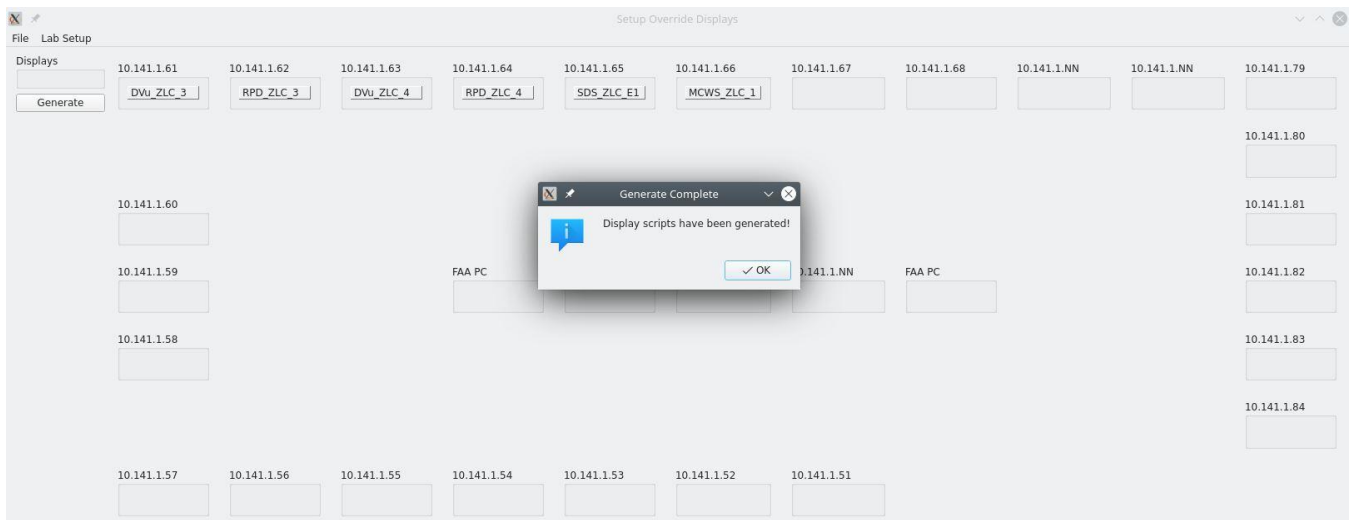


Figure 14. Configure ZTV Displays

Configure ZTV Displays

To configure ZTV displays, select **Tools** under each individual harnesses tab and click on “Configure ZTV Displays”. Then drag the displays onto your lab resources and then select generate.

4.1.6. SDRR Connector Configuration

The **sdrconnector** is used to connect multiple SDRR units to one another. The units need to be able to communicate with one another via a network and an sdrconnector configuration file must be specified. There is also an optional -g graphics mode which will display a status window of the connections that are

made and the messages between those connections. The SDRR connector configuration file can support ASCII formatted data by specifying format="ascii" within the interfacility device line.

Usage:

```
sdrconnector cfgFile [--logFile=logfile] [--guiMode] [--simDriverDev=device]
```

Example SDRR connector configuration file:

Sample config file:

```
<connections>
  <relay name="ppp-zny" isInterfacility="1" device1="pipe:ppp" device2="pipe:zny" />
  <relay name="cmsRelay" isInterfacility="0" device1="pipe:cms1" dev1Mode="readOnly"
device2="pipe:cms2" dev2Mode="writeOnly" />
  <relay name="serverRelay" isInterfacility="0" server1="localhost:7000"
device2="tcp://host2:8888" />
</connections>

<!-- old format -->
<connections>
  <connection name="PPP-ZNY" >
    <interfacility device="/dev/ifa0" txclock="0" rxclock="0" />
    <interfacility device="/dev/ifa1" txclock="0" rxclock="0" />
  </connection>
  <connection name="AAY-ZDC" >
    <interfacility device="ecgif:pipa/AAY" />
    <interfacility device="sirs5:/dev/ifa1" txclock="0" rxclock="0" />
  </connection>
</connections>
</connections>
```

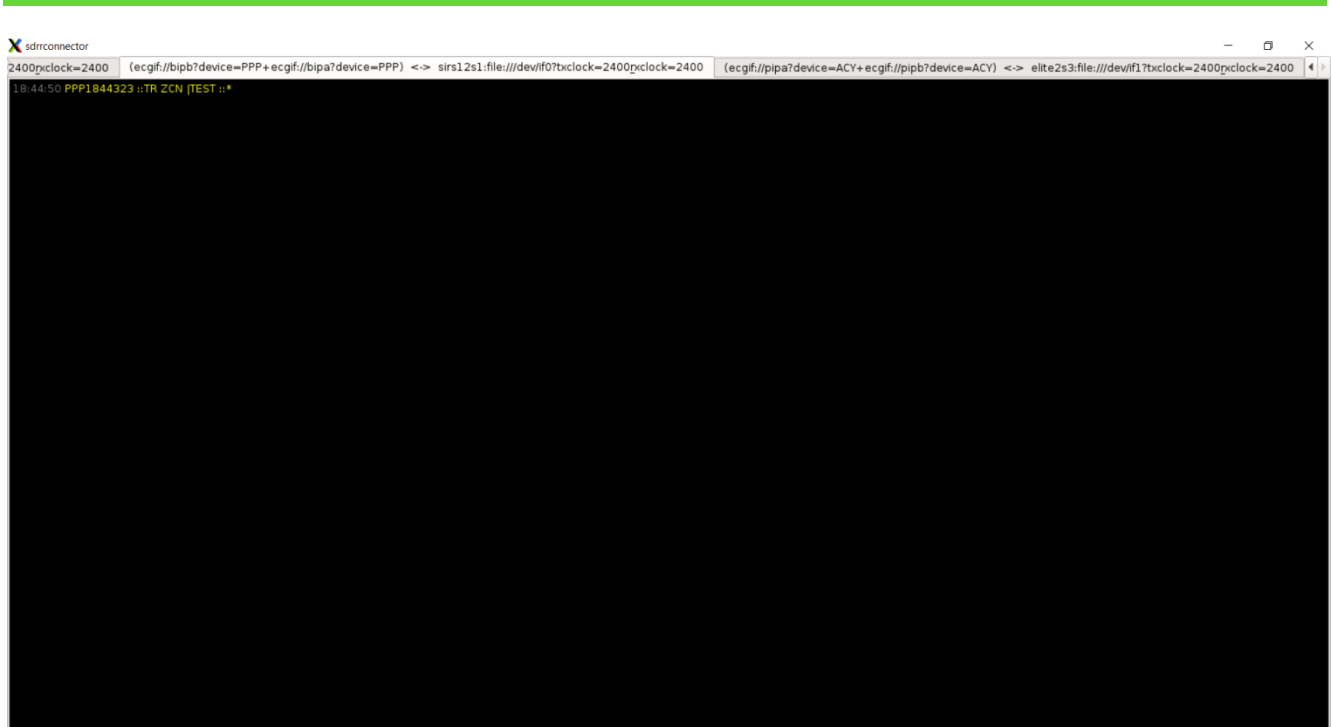


Figure 15. SDRR Connector

4.2. STARS CDR Utility

4.2.1. GUI Based Utility – STARSCDR

The GUI based STARS CDR utility extracts and plays back data from a STARS Continuous Data Recording (CDR) file. This utility parses this CDR data and creates a scenario based on the recorded data. The utility is launched by typing **STARSCDR** (case sensitive) at the command line:

> STARSCDR

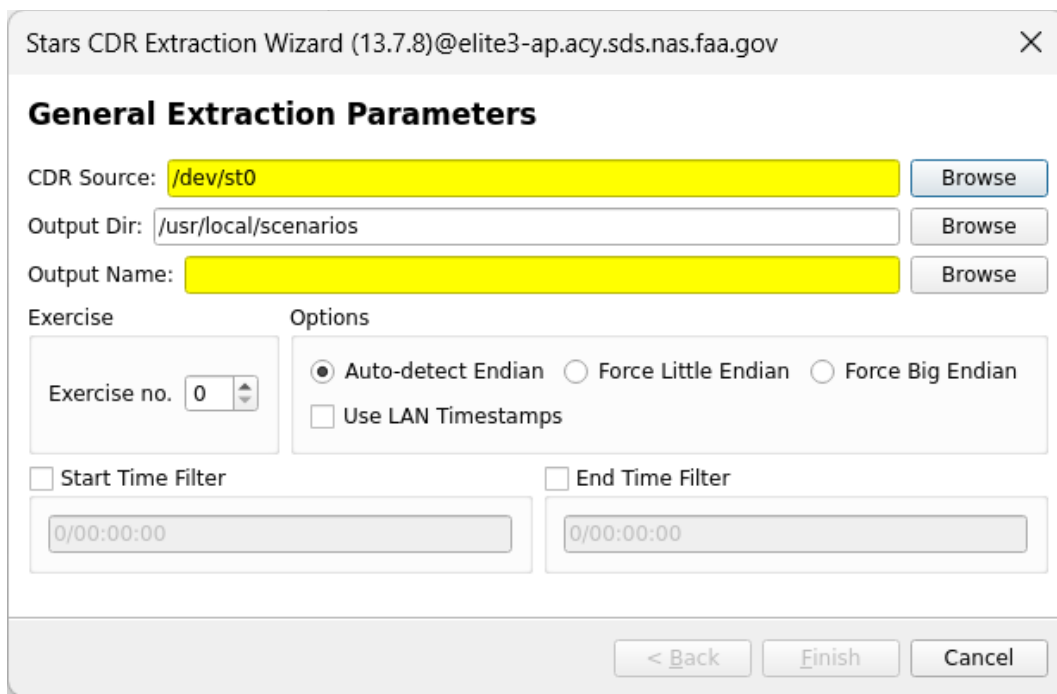


Figure 16. STARS CDR Extraction Wizard

CDR Source

Location of the CDR file. The default location is **/dev/st0** (tape drive device), but can also be on a CD or DVD. Users can also browse for a .cdr file by pressing the “Browse” button or typing the path in the text box.

Output Scenario Directory

Directory where the extracted scenario is saved. The default is the directory specified by the environment variable `${SDRR_SCENARIO_PATH}`.

Exercise no.

This refers to the mode STARS was in when the CDR file was recorded. The default exercise number is zero, indicating normal operation or live mode. If the CDR recording was done with STARS in "TEST MODE" or while ATCoach was running, the exercise number may need to be changed to 1 or 2.

Options

Auto Detect Endian – Determines which format, little Endian or Big Endian, is needed to extract the data from the CDR.

Force little Endian – Forces the extraction of the CDR using the little endian format.

Force Big Endian – Forces the extraction of the CDR using the big endian format.

Use LAN Timestamps – Uses LAN timestamps instead of default for exported serial radar.

Start Time Filter and End Time Filter

These options can be used to filter on certain times of interest in CDR data. Instead of extracting a 24 hour CDR, you may only need the first 30 minutes; you can set the end time to 30 and create a small SDRR scenario.

Finish

The finish button is pressed to extract the CDR.

After the data is extracted, the window below appears, in GREEN, (Success) or RED (Failed). Selecting the Details button provides details, such as Rev# site ID and message count to provide details on why an extraction may have failed.

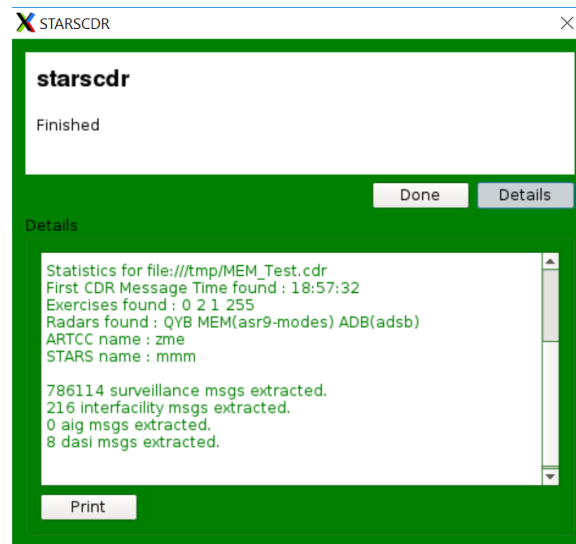


Figure 17. Extraction Successful Status

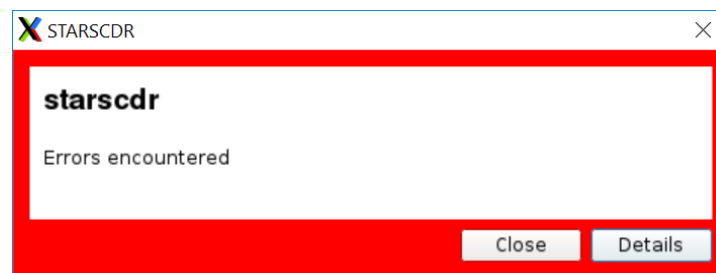


Figure 18. Extraction Error Status

4.2.2. Command Line Utility – starscdr

The starscdr command line utility is used to extract surveillance data from a STARS CDR tape or file.

```
> starscdr
```

```
Extract data from a STARS CDR tape/file.
JVN Communications Version 13.7.8
If no srv file is specified, info is printed.
Usage: starscdr cdrfile [-d outdir] [-e ex] [-a aigMsgFile (AIG200)] [-A aigMsgFile (AIG210)] [-k] [-s starttime]
[-f endtime] [-L] [-B] [-l] [-t] [-u]
    -L      force data to be read as little endian
    -B      force data to be read as big endian
    -l      list nonRadarMsgs to stdout
    -t      track list
    -u      extract TUs
    -k      extract keystrokes and store in 'keys.xml'
    -h      use lan header timestamps for Radar sensor messages
```

Figure 19: Command Line Utility starscdr

Parameter	Description
-d /output_directory	Specify an output directory path.
-e	Filter on specified exercises.
-D	By default, dasiMsgs.dasi files are auto generated.
-a	Output an aigMsgs file for AIG200 messages.
-A	Output an aigMsgs file for AIG210 messages.
-k filename.xml	Generate an xml file for playback keystrokes from the MCW. Note: User will have to move the generated keystroke XML over to CDR scenario.
-s hh:mm:ss	Specify a start time to filter CDR data (hh=hours, mm=minutes, ss=seconds).
-f hh:mm:ss	Specify an end time to filter CDR data (hh=hours, mm=minutes, ss=seconds).
-L	Force the extraction of the CDR using little endian format.
-B	Force the extraction of the CDR using big endian format.
-l	List nonRadarMsgs to stdout.
-t	List the target's tracks information.
-u	Extract track update (TU) positions. By default, starscdr extracts FP and AM messages only.
-k	Extract keystrokes and store in 'keys.xml'.

Parameter	Description
-h	Use LAN timestamps for extracted serial radar devices.

The example below, generates a CDR scenario to the specified output directory. This utility detects which STARS software version the CDR was created with and lists the statistics after extraction.

```
> starscdr /input_cdrfile.cdr -d /output_directory
```

```
# starscdr /usr/local/scenarios/CDR_Files/PPP.09SEP2023_211500.cdr -d /usr/local/scenarios/PPP.09SEP2023.06202024
STARS Software Version s6 r9.14
Using block size: 0x1f4000
File 0 size: 23702016 (0x169aa00)
21:15:00
22:14:59

Done Extraction.

Statistics for /usr/local/scenarios/CDR_Files/PPP.09SEP2023_211500.cdr
First CDR Message Time found : 21:15:00
Exercises found : 0 255
Radars found : HPN(asr9-modes) JFK(asr9-modes) ABE(asr11) PHL(asr9-modes) WRI(asr11) QIE(arsr4) NXY(asr11) ISP(asr9-modes) EWR(asr9-modes)
ADS(adsb) DOV(asr11) SWF(asr9-modes)
ARTCC name : zcn
STARS name : ppp

1403376 surveillance msgs extracted.
185 interfacility msgs extracted.
7138 aig msgs extracted.
8997 dasi msgs extracted.
```

4.3. Surveillance Data Utilities

4.3.1. SRV/DASR to Tracks

The `srv2tracks` and `dasr2tracks` utilities both provide track listings for the selected data types. For example, these utilities can be used to track the location of a specific aircraft by its beacon code. Enter the utility name, **`srv2tracks`** or **`dasr2tracks`**, for a listing of command options.

```
> srv2tracks
```

Or

```
> dasr2tracks
```

```
sdrr@dogfish:~$ srv2tracks
No configFile specified.
Usage: srv2tracks [--srvFile=file || --comdigFile=file] --configFile=file --radar=name [--psr] [--bcn=xxxx --bcn=yyyy...]

sdrr@dogfish:~$ dasr2tracks
No configFile specified.
Usage: dasr2tracks [--dasrFile=file || --comdigFile=file] --configFile=file --radar=name [--testTgts] [--psr] [--bcn=xxxx --bcn=yyyy...]
```

Figure 20: Utilities `srv2tracks` and `dasr2tracks`

The example below tracks the beacon code 4444 on `phl.srv`:

```
> srv2tracks --srvFile=phl.srv --configFile=phl.xml --radar=PHL --bcn=4444
```

The example below tracks the beacon code 4444 on `dox.ast`:

```
> dasr2tracks --srvFile=dox.ast --configFile=phl.xml --radar=DOX --bcn=4444
```

Other available options included are to specify comdig file inputs (`--comdigFile=file`), include primary scan reports (`--psr`), and `dasr2tracks` allows for the ability to include test targets (`--testTgts`).

When utilizing the utility please note that three things are needed:

1. an input radar file or comdig file

2. a site config file
3. a specific radar.

The following example was generated using the command:

```
> srv2tracks --srvFile=ph1.srv --configFile=ph1-regression.xml --radar=PHL --bcn=4001
```

```
<trajectoryList>
  <trajectory name="BCN4001">
    <update alt="10000.0" hdg="0.0" spos="+39:48:28.64, -075:33:28.62" t="3.70" tas="0.0"/>
    <update alt="9900.0" hdg="77.1" spos="+39:48:30.51, -075:33:18.02" t="8.32" tas="108.8"/>
    <update alt="9800.0" hdg="77.1" spos="+39:48:32.18, -075:33:08.63" t="12.94" tas="96.5"/>
    <update alt="9700.0" hdg="77.1" spos="+39:48:33.84, -075:32:59.23" t="17.56" tas="96.5"/>
    <update alt="9600.0" hdg="67.7" spos="+39:48:36.70, -075:32:50.18" t="22.18" tas="97.8"/>
    <update alt="9600.0" hdg="77.2" spos="+39:48:38.38, -075:32:40.63" t="26.80" tas="98.1"/>
    <update alt="9500.0" hdg="77.2" spos="+39:48:40.25, -075:32:30.03" t="31.42" tas="108.8"/>
    <update alt="9500.0" hdg="77.2" spos="+39:48:41.93, -075:32:20.48" t="36.04" tas="98.1"/>
    <update alt="9400.0" hdg="77.2" spos="+39:48:43.58, -075:32:11.07" t="40.66" tas="96.5"/>
    <update alt="9400.0" hdg="68.4" spos="+39:48:46.40, -075:32:01.85" t="45.28" tas="99.3"/>
    <update alt="9400.0" hdg="77.2" spos="+39:48:48.07, -075:31:52.29" t="49.90" tas="98.1"/>
    <update alt="9300.0" hdg="77.3" spos="+39:48:49.71, -075:31:42.88" t="54.52" tas="96.5"/>
    <update alt="9300.0" hdg="77.3" spos="+39:48:51.38, -075:31:33.32" t="59.14" tas="98.1"/>
    <update alt="9200.0" hdg="69.7" spos="+39:48:54.32, -075:31:23.03" t="63.76" tas="109.8"/>
    <update alt="9200.0" hdg="77.3" spos="+39:48:55.97, -075:31:13.46" t="68.38" tas="98.1"/>
    <update alt="9100.0" hdg="77.3" spos="+39:48:57.60, -075:31:04.05" t="73.00" tas="96.5"/>
    <update alt="9100.0" hdg="77.3" spos="+39:48:59.26, -075:30:54.48" t="77.63" tas="97.9"/>
    <update alt="9100.0" hdg="69.3" spos="+39:49:01.96, -075:30:45.21" t="82.25" tas="99.2"/>
    <update alt="9000.0" hdg="77.4" spos="+39:49:03.58, -075:30:35.80" t="86.87" tas="96.5"/>
    <update alt="8900.0" hdg="77.4" spos="+39:49:05.20, -075:30:26.39" t="91.49" tas="96.5"/>
    <update alt="8900.0" hdg="69.6" spos="+39:49:07.86, -075:30:17.10" t="96.11" tas="99.1"/>
    <update alt="8800.0" hdg="77.5" spos="+39:49:09.46, -075:30:07.68" t="100.73" tas="96.5"/>
    <update alt="8700.0" hdg="77.5" spos="+39:49:11.07, -075:29:58.26" t="105.35" tas="96.5"/>
    <update alt="8600.0" hdg="69.9" spos="+39:49:13.65, -075:29:49.13" t="109.97" tas="97.4"/>
    <update alt="8500.0" hdg="77.6" spos="+39:49:15.45, -075:29:38.51" t="114.59" tas="108.8"/>
    <update alt="8400.0" hdg="77.6" spos="+39:49:17.04, -075:29:29.09" t="119.21" tas="96.5"/>
    <update alt="8300.0" hdg="77.6" spos="+39:49:18.64, -075:29:19.66" t="123.83" tas="96.5"/>
    <update alt="8200.0" hdg="70.3" spos="+39:49:21.16, -075:29:10.51" t="128.45" tas="97.3"/>
    <update alt="8200.0" hdg="77.7" spos="+39:49:22.77, -075:29:00.92" t="133.07" tas="98.2"/>
    <update alt="8100.0" hdg="70.6" spos="+39:49:25.27, -075:28:51.75" t="137.69" tas="97.3"/>
    <update alt="8000.0" hdg="77.8" spos="+39:49:26.84, -075:28:42.32" t="142.31" tas="96.5"/>
    <update alt="7900.0" hdg="77.8" spos="+39:49:28.41, -075:28:32.89" t="146.93" tas="96.5"/>
    <update alt="7800.0" hdg="70.9" spos="+39:49:30.86, -075:28:23.71" t="151.55" tas="97.2"/>
    <update alt="7700.0" hdg="77.9" spos="+39:49:32.42, -075:28:14.27" t="156.17" tas="96.5"/>
    <update alt="7600.0" hdg="77.9" spos="+39:49:33.98, -075:28:04.84" t="160.79" tas="96.5"/>
    <update alt="7500.0" hdg="71.3" spos="+39:49:36.38, -075:27:55.64" t="165.41" tas="97.2"/>
    <update alt="7500.0" hdg="78.0" spos="+39:49:37.95, -075:27:46.04" t="170.03" tas="98.2"/>
    <update alt="7400.0" hdg="71.5" spos="+39:49:40.32, -075:27:36.83" t="174.65" tas="97.1"/>
    <update alt="7300.0" hdg="78.1" spos="+39:49:41.86, -075:27:27.39" t="179.27" tas="96.5"/>
  </trajectory>
</trajectoryList>
```

Figure 21. Example of srv2tracks Output

4.3.2. Track Flagging Utilities

The track flagging utilities are used to flag certain tracks by beacon code or other target attributes. Once the data file contains flagged (bit enabled) targets, SDRR sends a signal to the machine's parallel port every time the target is processed, once per scan. This is used for Target to Glass performance testing to measure any latency between process time and delivery time. With an LED attached to the parallel port of SDRR, precise measurements can be taken between process and glass time.

The utilities are launched by typing the utility name **astmsgflag**, **azflag**, **bcnflag**, or **setflag** at the command line:

```
> astmsgflag
```

```
> azflag
```

```
> bcnflag
```

```
> setflag
```

```
sdrr@troy:~ #astmsgflag
JVN Communications, 2008
Set/clear the parallel port trigger flag in messages in ASTERIX file. Version 1.4 2013/10/24
Requires in_file, out_file, and one of -a, -i, -b, or -m
Usage: astmsgflag in_file out_file [-r] [-q] [-c <asterix_category>] [-s <start_time>] [-e <end_time>] [-a|-i <icao_code>|-
b <bcn_code>|-m <modes_id>]
-r : Clear the parallel port trigger flag instead of setting.
-q : Expect DASR sequence number.
-c <asterix_category> : Only apply the set/clear operation on category <asterix_category> messages. Defaults to all.
-a : Apply the set/clear operation on ALL messages of the selected ASTERIX category.
-i <icao_code> : Only apply the set/clear operation on messages containing an ICAO code field that is set to <icao_c
ode>.
-b <bcn_code> : Only apply the set/clear operation on messages containing a mode 3/A code field that is set to <bcn
_code>.
-m <modes_id> : Only apply the set/clear operation on messages containing a mode S ID field that is set to <modes_i
d>.
-s <start_time> : Only apply the set/clear operation on messages after <start_time>.
-e <end_time> : Only apply the set/clear operation on messages before <end_time>.
-n <interval> : Only apply the set/clear operation every <interval> messages.

<asterix_category> is in decimal format.
<icao_code> is in hexadecimal format.
<bcn_code> is in octal format.
<modes_id> is a text string.
sdrr@troy:~ #
```

Figure 22. ASTERIX Message Flagging Utility

4.3.3. Change SRV Utilities

The utilities **chalt**, **chgbcn**, **chgchan**, and **chgtime** manipulate SDRR SRV files (.srv) in the following ways:
chgalt – changes a targets altitude for the duration of the specified SRV file

```
> chgalt
```

chgbcn – changes the beacon code, of a target, for user defined durations

```
> chgbcn
```

chgchan – changes the recorded channel number in the radar file to a user defined channel

```
> chgchan
```

chgtime – changes the start time of an SRV file

```
> chgtime
```

Type the utility name for a list of command options.

```
jason@madelf:~$ chg
chgStatusBits  chgadsbvalidity  chgastalt      chgbcn         chgcpc         chgmodes       chgrdrd       chgtime
chgacp         chgalt          chgasttime    chgchan        chgfrn         chgnicnacsil  chgrp
jason@madelf:~$ chgalt
Usage: chgalt infile outfile -c modec [-b bcn] [-s starttime] [-e endtime]
jason@madelf:~$ chgbcn
2005 JVN Communications, Inc.
Change a beacon code
Usage : chgbcn infile outfile orig_beacon new_beacon [start_time [stop_time]]
jason@madelf:~$ chgchan
Usage: chgchan infile outfile [chan,(chan,chan,...)]
       default chan is 0
jason@madelf:~$ chgtime
Usage: chgtime infile outfile [dt]
       if no dt specified, outfile will be adjusted to start at 0
jason@madelf:~$
```

Figure 23. Change SRV Utilities

4.3.4. Strip Utilities

The utilities **stripbcn**, **stripastbcn**, **stripchan**, **stripAdsRs**, and **stripwx** manipulate SDRR surveillance files in the following ways:

stripbcn – removes messages for one or more beacon codes from an SRV file or removes all messages except those for beacon codes specified by the -k (keep) option. Messages from the entire file are removed unless start and end times are specified.

```
> stripastbcn --help
```

```
Usage: stripastbcn infile outfile -b bcn1,bcn2,bcn3... [-s starttime] [-e endtime] [-q]
```

```
-q    skip dasr seqnos
```

```
-S    not from SBS (for ASDE-X ASDEX)
```

```
-k    keep specified bcn(s) and strip all others
```

```
Strips the messages with specified bcn. Strips messages from entire file if start/end not specified.
```

```
otherwise only messages within specified timeframe are removed.
```

stripastbcn – removes messages for one or more beacon codes from an AST file or removes all messages except those for beacon codes specified by the -k (keep) option. Messages from the entire file are removed unless start and end times are specified.

```
> stripbcn --help
```

```
Strip (or keep) beacon code(s) from a srv file.
```

```
Usage : stripbcn infile outfile -b bcn1,bcn2,bcn3... [-s start_time [-e stop_time]] [-k]
```

```
Multiple bcns may be specified but must be comma separated.
```

```
If specified, only messages between chosen start and end times will be stripped.
```

stripchan – removes messages for the specified channel from an SRV file. Messages from the entire file are removed unless start and end times are specified.

```
> stripchan -help
```

```
Usage: stripchan inFile [-c chan,(chan,chan,...)] -o outFile
```

stripAdsbRs – removes messages for the specified Radio Station LID from an ADSB file containing Cat33 messages. Messages from the entire file are removed unless start and end times are specified.

```
> stripAdsbRs --help
```

```
Usage: stripAdsbRs infile outfile -l lid1,lid2,lid3... [-s starttime] [-e endtime] [-q]
Strips the messages with specified RadioStation LID. Strips messages from entire file if
start/end not specified.
otherwise only messages within specified timeframe are removed.
!This utility is meant to be used on ADSB files containing Cat33 messages!
```

stripwx – removes weather messages from an SRV file, or removes all messages except for weather if the -w option is specified. Messages from the entire file are removed unless start and end times are specified.

```
> stripwx --help
```

```
Usage: inFile outFile [-w] [-s startTime] [-e endTime]
-w  create file with wx msgs only
-s  only remove wx (or everything else with '-w' option) after specified start time
-e  only remove wx (or everything else with '-w' option) before specified end time
```

4.3.5. Find Utilities

The find utilities, **findtime** and **findstatic**, are used on SRV files as analysis tools.

findtime – shows the start and end time of an SRV file

```
> findtime
```

findstatic - locates test targets, such as: PARROTS, PE, SRTQC, and BRTQC, in radar data

```
> findstatic
```

The utility names can each be entered at the command line for a list of options.

```
# findstatic --help
findstatic: invalid option -- '-'
Copyright (c) Mike Natale, JVN Communications, Inc.
Find "static" messages in a surveillance file.
Usage : findstatic radar_file [-z num] [-r num] [-a num] [-h num] [-m num]
      [-n num] [-p per] [-s time] [-e time] [-d] [-S] [-I] [-E] [-P] [-M] [-A] [-t] [-y]
radar_file : Surveillance file in question.
-z <num>    : Allow <num> ACPs deviance in azimuth values.(12)
-r <num>    : Allow <num> deviance in range values.(16)
-a <num>    : Allow <num> 100s ft deviance in altitude values.(5)
-h <num>    : <num> hits makes a message "static".(-1)
-m <num>    : <num> scans w/o hit to forget a message.(3)
-n <num>    : Run through only <num> scans instead of whole file.(20)
-p <num>    : Percent of scans to make a message "static".(75.0 %)
-s <time>   : Start at HH:MM:SS instead of BOF.
-e <time>   : Stop at HH:MM:SS instead of EOF.
-x <scale>  : Interpret the range field units to be 1/scale Nmi.(64)
-d          : Print out LOTS of debug info.
-D          : Print degrees instead of ACP in English mode.
-l          : Long range format printout.
-S          : Allow NO deviance in any fields.
-I          : Allow no deviance in fields 1, 4, and 6 of any BCN/SRC msg.
-E          : Print messages in plain English.
-P          : Show percentage of hits instead of start and stop times.
-M          : Don't print sector mark messages.
-A          : Print min/mean/median/max values.
-t          : Print to match STARS reports.
-y          : Treat input as Asterix format
-q          : Skip sequence messages when parsing data (DASR)
```

Figure 24.Command Line Options for Find Utilities

The **findstatic** utility is also available in a GUI version. The **FindStatic** GUI allows multiple radar files to be scanned simultaneously and the output saved to a file. When executing on multiple radars at once, ensure the radars are all the same type, i.e., short range or long range. Select the Long Range box for long range radar. To launch the GUI, **FindStatic** can be entered at the command line.

```
> FindStatic
```

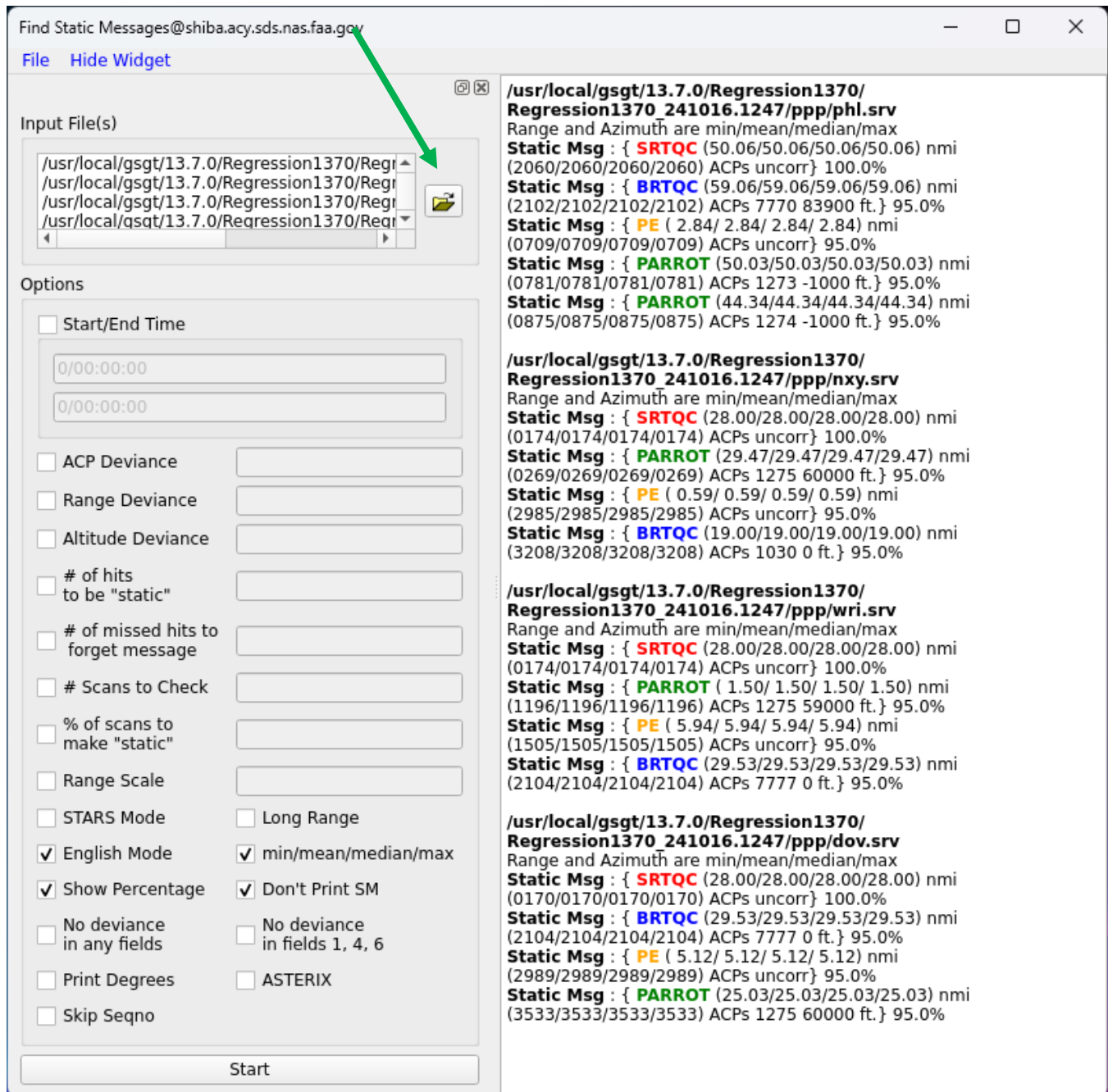


Figure 25. Find Static Messages GUI

4.3.6. Loop SRV

The Loop SRV utility, **loopsrv**, is used to extend the length of SRV files. This utility copies the original SRV file and appends it to the end of the file to extend the duration. The utility name, **loopsrv**, can be entered at the command line for a list of options.

```
> loopsrv
```

```
jason@madelf:/usr/local/jvn.11.2.8/bin$ loopsrv  
loopsrv infile outfile [-n reps] [-t looptime(hundredths)]  
jason@madelf:/usr/local/jvn.11.2.8/bin$
```

Figure 26. Loop SRV Options

4.3.7. Merge Utilities

The merge utilities, **mergeJVNFiles** and **mergeSRVFiles**, are used to merge JVN and SRV files.

mergeJVNFiles - merges two JVN interfactivity files (*.jvn) into a single JVN interfactivity file.

The utility name can be entered at the command line to view the options.

```
> mergeJVNFiles
```

```
jason@madelf:/usr/local/scenarios/PHL$ mergeJVNFiles  
Usage: mergeJVNFiles [infile1] [infile2] -o [outFile]  
This utility merges 2 JVNFFiles and outputs the sorted data  
to the specified file  
jason@madelf:/usr/local/scenarios/PHL$ mergeJVNFiles 04282020.0000/zny-ppp.jvn 04292020.0000/zny-ppp.jvn -o zny-ppp-merged.jvn
```

Figure 27. Merge JVN Files

mergeSRVFiles - merges two or more SRV radar files (*.srv) into a single SRV file.

The utility name can be entered at the command line to view the options.

```
> mergeSRVFiles
```

```
jason@madelf:/usr/local/scenarios/PHL$ mergeSRVFiles
Usage: mergeSRVFiles inFile [...inFile] -o outFile
This utility merges srvFiles and outputs the sorted data
to the specified file
jason@madelf:/usr/local/scenarios/PHL$ mergeSRVFiles 04282020.0000/phl.srv 04292020.0000/phl.srv -o phl-merged.srv
```

Figure 28. Merge SRV Files

4.3.8. CD2 Conversion

The CD2 conversion utility converts CD2 format data including CD2 weather to the SDRR SRV data format. The utility can be executed entirely from the command line with the **cd2srv** executable.

```
> cd2srv
```

Or, a GUI can be launched with the **cd2convert** executable.

```
> cd2convert
```

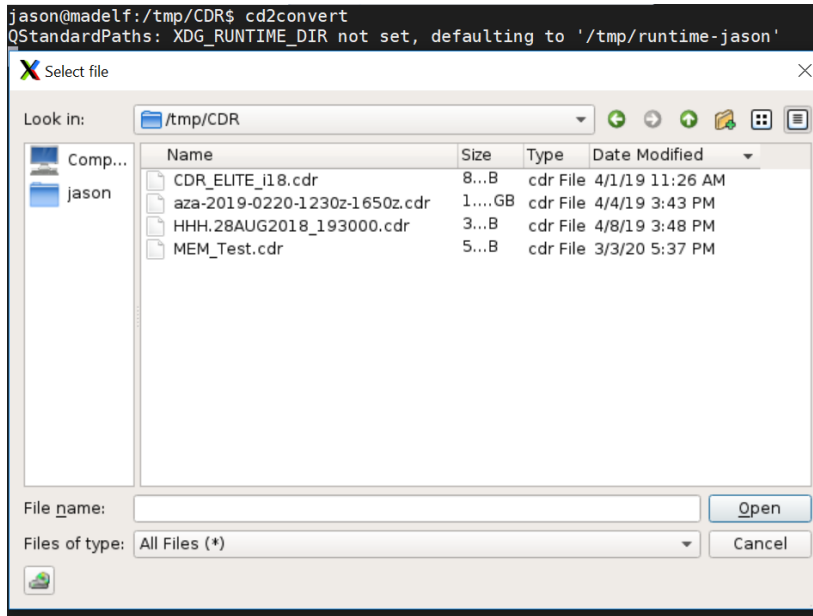


Figure 29. CD2 Conversion GUI

4.3.9. Scan Time

The **scantime** utility is used with surveillance (.srv) files to determine a radar's scan time. Most ASR9 radar are discovered in approximately 4.6 seconds. The utility name can be entered at the command line for a list of options.

```
> scantime
```

```
jason@madelf:~$ scantime
Calculate the scan time of an .srv file.
JVN Communications, Inc. 2003, Version 1.5 2016/12/30
Usage : scantime <srv_file/device> [-r]
        -r Give real-time stats for files.
jason@madelf:~$
```

Figure 30. Scan Time Options

4.3.10. Time Slice

The Time Slice utility pulls a section of radar data from a larger file. For example, if the first thirty minutes are needed from a ten hour scenario, the Time Slice utility can be used to cut out and rename the first thirty minute section. This allows a specific time frame to be examined without loading a lengthy scenario. Type the utility name, **timeslice**, for a list of command options.

```
> timeslice
```

```
jason@madelf:~$ timeslice
Usage: timeslice -t startTime,endTime infile outfile
jason@madelf:~$
```

Figure 31. Time Slice Utility

The example below, limits the phl.srv file time to the first thirty minutes using timeslice command:

```
> timeslice -t 00:00:00,00:30:00 phl.srv phl-00-30minutes.srv
```

The Time Slice utility is also available in a GUI version that allows users to cut an SDRR scenario into slices by specifying the start and end times of the slice. This utility can be launched either from an icon on the desktop (if created) or by typing **TimeSlice** (case sensitive) in a terminal window. This will launch the TimeSlice GUI interface:

```
> TimeSlice
```

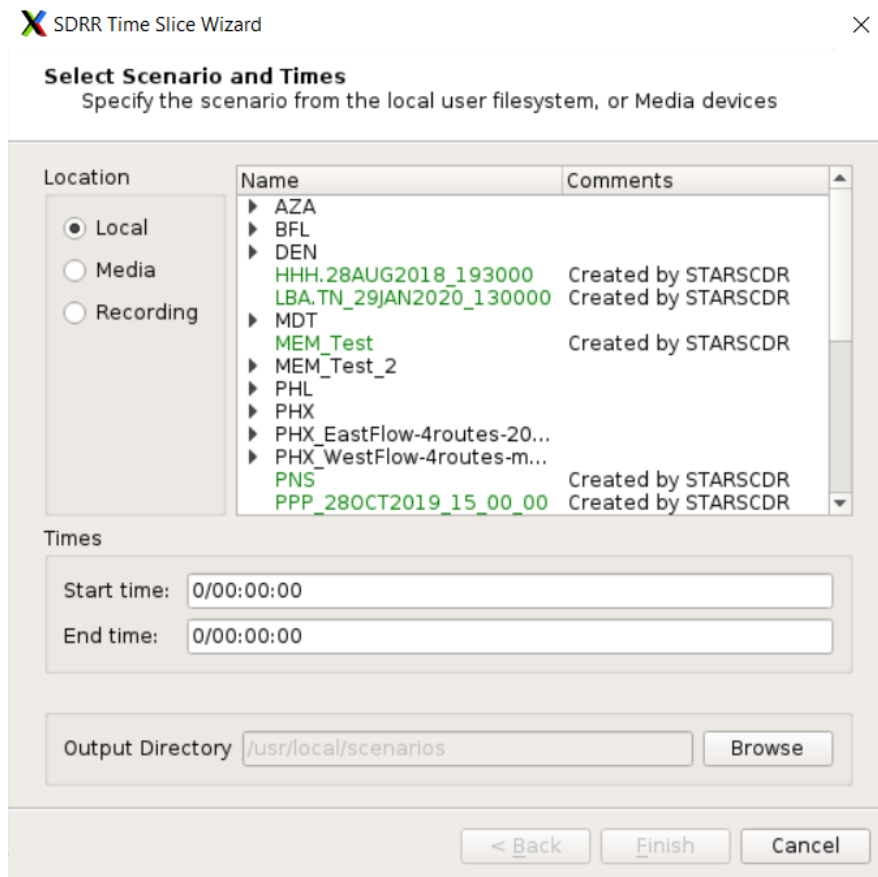
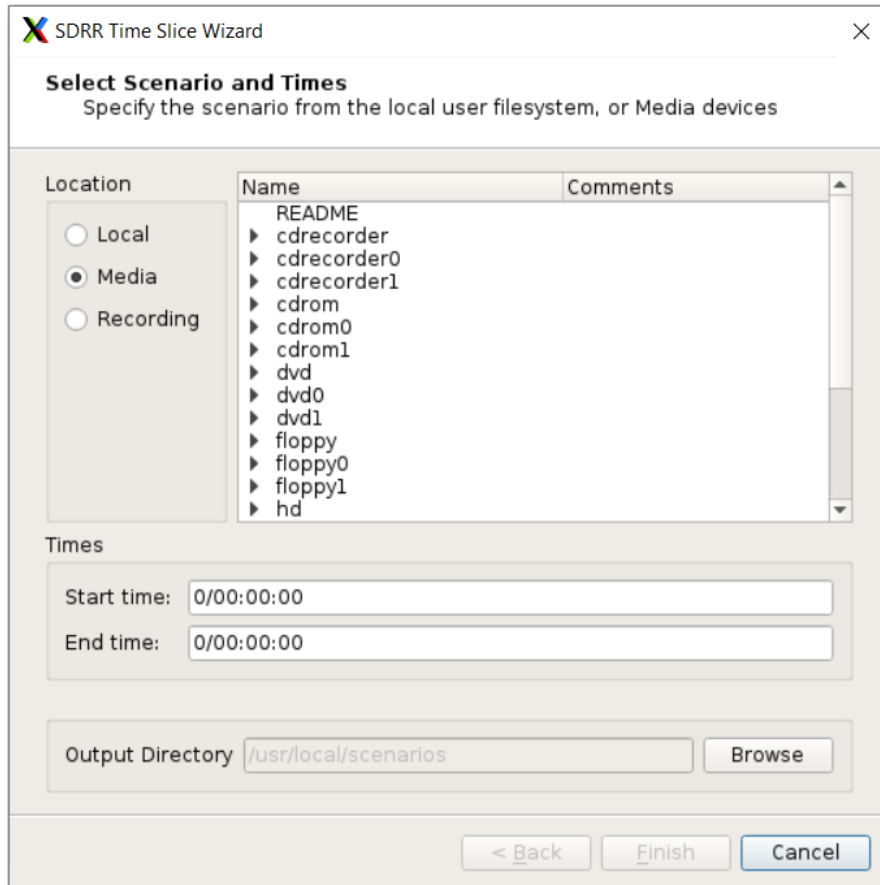


Figure 32. Time Slice GUI

When the utility is launched, it defaults to the local user file system directory specified by the environment variable `${SDRR_SCENARIO_PATH}`, but the user can also select a scenario from a media device.



SDRR Time Slice Wizard

Select Scenario and Times
Specify the scenario from the local user filesystem, or Media devices

Location

☐ Local

☒ Media

☐ Recording

Name	Comments
▶ README	
▶ cdrecorder	
▶ cdrecorder0	
▶ cdrecorder1	
▶ cdrom	
▶ cdrom0	
▶ cdrom1	
▶ dvd	
▶ dvd0	
▶ dvd1	
▶ floppy	
▶ floppy0	
▶ floppy1	
▶ hd	

Times

Start time: 0/00:00:00

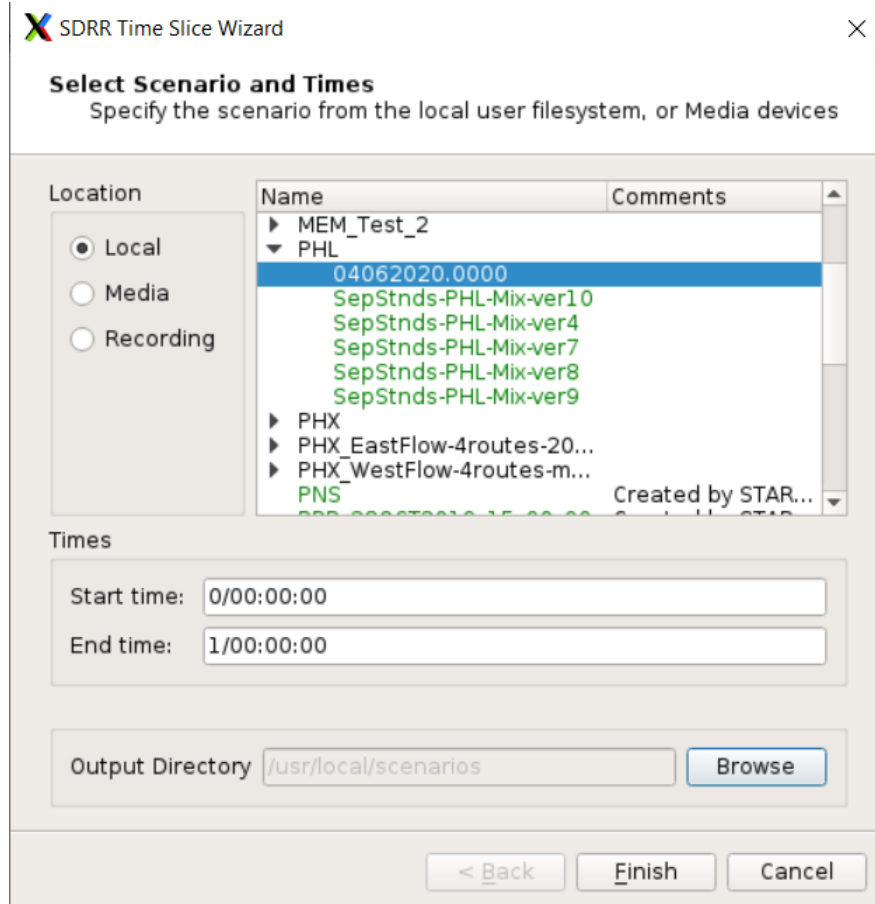
End time: 0/00:00:00

Output Directory: /usr/local/scenarios Browse

< Back Finish Cancel

Figure 33. Selecting a Media Device

When the user selects a scenario, the start time and end time of the scenario is displayed in the format D/HH:MM:SS (where D=days, HH=hours, MM=minutes, SS=seconds).



SDRR Time Slice Wizard

Select Scenario and Times
Specify the scenario from the local user filesystem, or Media devices

Location

- ☒ Local
- ☐ Media
- ☐ Recording

Name	Comments
MEM_Test_2	
PHL	
04062020.0000	
SepStnds-PHL-Mix-ver10	
SepStnds-PHL-Mix-ver4	
SepStnds-PHL-Mix-ver7	
SepStnds-PHL-Mix-ver8	
SepStnds-PHL-Mix-ver9	
PHX	
PHX_EastFlow-4routes-20...	
PHX_WestFlow-4routes-m...	
PNS	Created by STAR...

Times

Start time: 0/00:00:00

End time: 1/00:00:00

Output Directory: /usr/local/scenarios Browse

< Back Finish Cancel

Figure 34. Start Time and End Time Auto-populated

The start time and/or the end time can be changed to narrow the scenario to the desired time. For example, if the event of interest occurred between 00:00:00 and 06:00:00, 0/00:00:00 can be entered in the start time box and 0/06:00:00 can be entered in the end time box. The “Finish” button can be pressed once the changes are made.

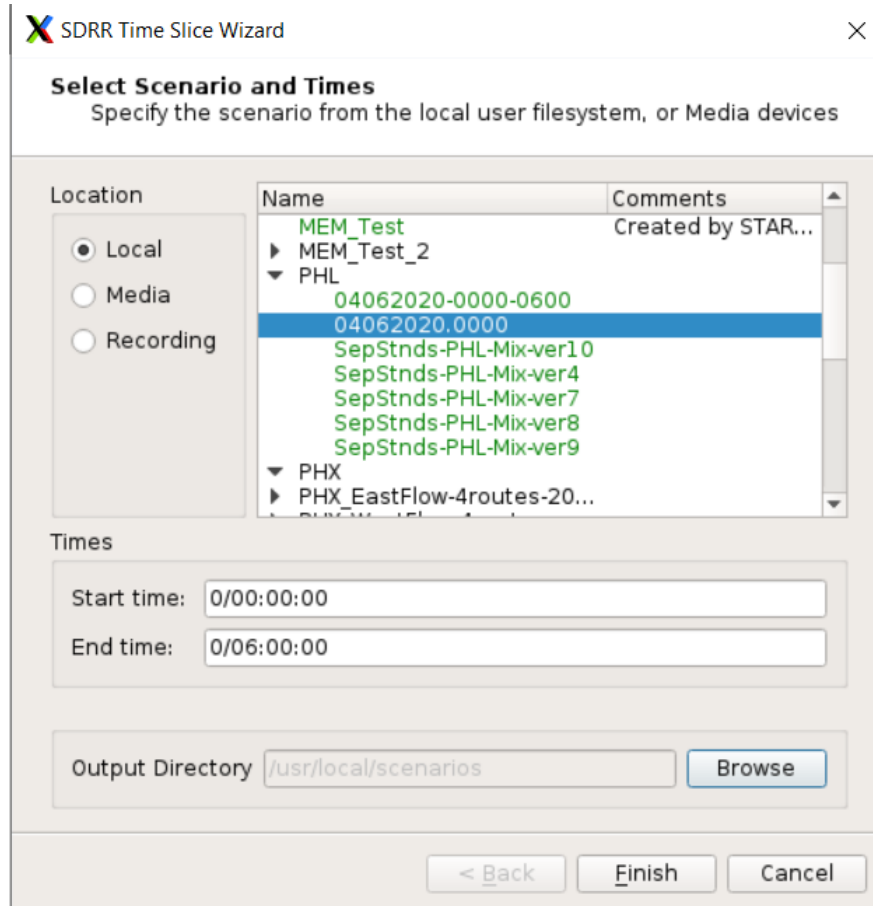


Figure 35. End Time Manually Changed

The execution status window displays the individual steps executed in order to slice the scenario. The status window will display “Finished” when the execution is completed.

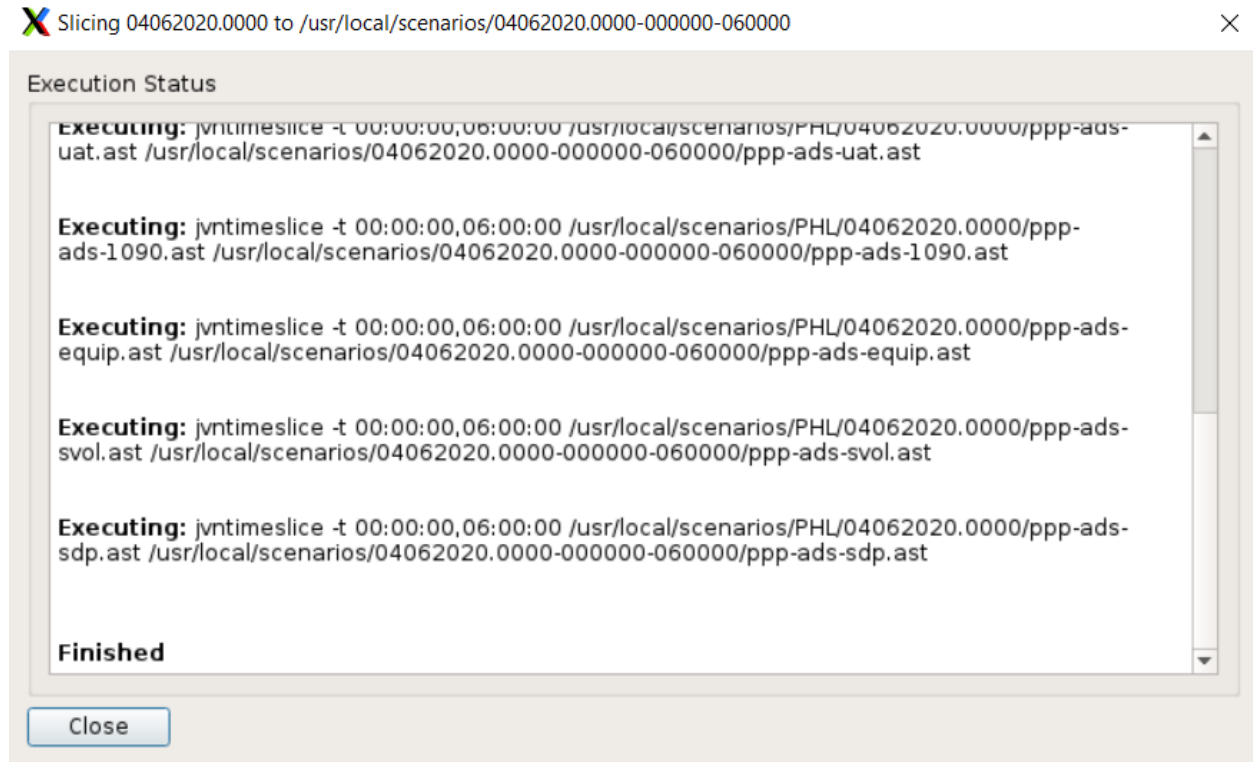


Figure 36. Execution Status Window

Pressing the “Close” button will exit the status window. To verify that the new slice was saved, the scenario directory can be examined or the Time Slice GUI can be re-opened. The new slice is saved with the original filename with the start time and end time appended to the end of the name.

SDRR Time Slice Wizard ×

Select Scenario and Times
Specify the scenario from the local user filesystem, or Media devices

Location

☒ Local

☐ Media

☐ Recording

Name	Comm
04062020-0000-0600-000000-060000	
04062020.0000-000000-060000	
▶ AZA	
▶ BFL	
▶ DEN	
HHH.28AUG2018_193000	Creat
LBA.TN_29JAN2020_130000	Creat
▶ MDT	
MEM_Test	Creat
▶ MEM_Test_2	
▶ PHL	

Times

Start time: 0/00:00:00

End time: 0/06:00:00

Output Directory: /usr/local/scenarios Browse

< Back Finish Cancel

Figure 37. New Sliced Scenario

4.4. Analysis Utilities

4.4.1. ADS-B Network Statistics Utility

This utility is a modified version of the open source network analyzing application, *Wireshark*, and displays statistical information from ADS-B targets. The utility is launched by typing **adsbpcapstats** at the command line:

```
> adsbpcapstats
```

```
jason@madelf:/tmp$ adsbpcapstats
JVN Communications, Inc. 2008
Print ADSB cat 33 field values (see below) from wireshark recording (PCAP).
Values in order : TOR (timestamp), TOA (4), TOMR (18), TOO (24), TOMR - TOMA, SVID (1), Ver Stat
(2), Version (2), Region Prefix (22), Equip Type (22), Loc ID (22), Instance (22), Link Type [A
DSB Detection (3)+ UAT (3)+ R1090 (3)], Link Ver (3), Dup Addr (5), Addr Qual (5), Tgt Addr (5),
Mode 3A Valid (11), Mode 3A Code (11), Position (7), Pressure Alt (8), Tgt ID (12), NIC Baro (6
), Validation (6), Test Mode (6), NACv (6), NACv Avail (6), NACp (6), NACp Avail (6), SIL (6), N
IC (6), UTC Coupled (6), Rpt ID (23), Lat (7), Lon (7)
Usage : adsbpcapstats <adsb_file> [-2|-3] [-23|-33]
      <adsb_file> can be pcap (wireshark, tcpdump) or .ast file
      -2 Process version 2 data. (Version 3 is default)
      -3 Process version 3 data. (default)
      -s Process CAT 23 messages. (CAT 33 is default)
      -t Process CAT 33 messages. (default)
jason@madelf:/tmp$
```

Figure 38. ADS-B Network Statistics

4.4.2. Network Capture Conversion

The **pcap2jvn** utility converts a Wireshark network capture (.cap) file to an SDRR (.ast) file. Once converted, the new file can be played back in SDRR or viewed with the **msglist** utility. The utility name can be entered at the command line for a list of options.

```
> pcap2jvn
```

```
jason@madelf:/usr/local/jvn.11.2.8/bin$ pcap2jvn

JVN Communications, Inc. 2008
Convert wireshark recording (PCAP) to jvn format
pcap2jvn pcapFile outFile [-q] [-h skipHeadPayloadBytes] [-m minDataSize] [-t skipTailPayloadBytes] [-0 "filter1"] [-1 "filter1"] .....
-q virtual network (skip 1st 2 bytes of each packet)

jvn msg will be flagged with 'n' representing the filter matched

example: pcap2jvn file.pcap tmp-uat.ast -0 "dst 239.161.14.32 and port 59950"
jason@madelf:/usr/local/jvn.11.2.8/bin$
```

Figure 39. Network Capture Conversion

4.4.3. Monitor

The **monitor** utility displays the physical radar and interfacility devices that are installed in the SDRR processor.

```
> monitor
```

```
sdr@elite2-master:~$ monitor elite2s3
Info from elite2s3:

Radars:

rdr0: 6:8/0, 6:8/1, 6:8/2, 6:8/3      unopened (ELITE2-/dev/srr0-NUN1-MOD4-A)
rdr1: 6:9/0, 6:9/1, 6:9/2, 6:9/3      unopened (ELITE2-/dev/srr1-NUN2-MOD1-A)
rdr2: 6:a/0, 6:a/1, 6:a/2, 6:a/3      unopened (ELITE3-/dev/srr2-NUN1-MOD4-A)
rdr3: 6:b/0, 6:b/1, 6:b/2, 6:b/3      unopened (ELITE3-/dev/srr3-NUN2-MOD1-A)

Interfacility:

if0: pcibus=6 slot=7 port=0 (ELITE2)
if1: pcibus=6 slot=7 port=1 (ELITE3)
if2: pcibus=6 slot=7 port=2 (SPARE)
if3: pcibus=6 slot=7 port=3 (SPARE)
```

Figure 40. Monitor Utility

4.4.4. Message List

The **msglist** utility is used to view SDRR radar SRV (.srv), asterix (.ast) or JVN interfacility (.jvn) files. The utility has many options to view the radar, asterix or interfacility data, including viewing in raw radar mode, (hex), and English mode, as well as searching on beacon targets and other aircraft attributes. In addition to viewing files, this utility can be used on physical devices. When used on a radar device, such as **/dev/srr0**, the utility assists in verifying if radar data is reaching SDRR.

The utility name can be entered at the command line for a list of options.

```
> msglist
```

```
Usage: msglist file [-b[bcncode]] [otherFilterOptions] [-s "starttime,endtime (starttime,endtime) ..."] [-c chan] [--stream=x]

default is srv decode. options:
  -e[range_scale] plain english decoding for ASR or any ASTERIX
  -E[range_scale] plain english decoding for ARSR or any ASTERIX
  use optional range_scale to interpret range field units as 1/range_scale NMi (16, 32, or 64)

  --jvn raw JVNMsg msg dump
  --hp raw HPMsg msg dump
  -I interfacility decode
  -A[012] ASTERIX decode
  --ecgp decode
  --cms decode
  --hgi decode
  --sip decode
  --ed137 decode
  --tcw decode
  --parsedCMS decode
  --parsedTraj decode
  --dis decode
  --aig decode (STARS AIG)
  --tbfm decode (TBFM AIG)

  -n print msgno
  -x print offset in file
  --invert inverts data [only works on real-time devices]

  --sbs skip BSDU header, crc present
  --dasr skip seqno, no crc
  --sim skip seqno, crc present

Other filter options:
  -r[bs]
  use optional argument 'b' for brtqcs, 's' for srtqcs. No args for both
  -i siteid msgs
  -y skip control msgs
  -w wx msgs
  -t strobe msgs
  -u status msgs
  -p primary (search) msgs
  -m sector_marks
  -a aims msgs
  -k trkno (Asterix)
  -C category (Asterix)
  -S filter on address (ICAO address for cat 33, ModeS Addr for cat 10)
  -T filter on target ID (ADSB)
  -F omit ADSB "0xfaafaa" static msgs
  -F omit AVLC broadcast messages

  --color, --nocolor (default is to autodetect based on stdout file type)
  --syntax display allowable pdev syntax
```

Figure 41. Message List Options

By default, the **msglist** utility writes the output to the terminal window. This may not be desired for viewing large amounts of data. Instead, the output can be directed to a file:

```
> msglist file [options] > /tmp/your_file_name
```

This format will redirect the output to the file '/tmp/your_file_name' and can be viewed using vim or other text editing applications.

Decoding surveillance (.srv) files

The following options are available when decoding surveillance files:

- -E : plain English decoding for ARSR or ASTERIX
- -c # : decode a particular radar channel
- -r[rb] : decode only SRTQC and BRTQC or both
- -b#### : decode all messages with particular beacon code

Decoding ADS-B or ASTERIX (.ast) files

The following options are available when decoding ADS-B and ASTERIX files:

- -E : plain English decoding for ARSR or ASTERIX
- -A : allows for decoding of ASTERIX files (required)
- -F : omit ADSB "0xfaafaa" static messages

Decoding interfacility messages (.ast) files

The following options are available when decoding interfacility :

- -E : plain English decoding for ARSR or ASTERIX
- -I : decodes interfacility message files

4.4.5. Filter JVn Log

The **filterJvnLog** utility lets users extract complete stanzas containing specified text from XML log files. Stanzas can be extracted because they include text specified with the `--filter` option or because they do not include text specified with the `--discard` option.

```
> filterJvnLog --help
```

```
Usage: filterJvnLog [--filter="text"]... [--discard="text"]...  [--navcanFilter=aid] [--nonLogFile] logFile [...logFile]
```

Multiple filter options will be processed as an "OR" when filtering

`--filter="FU111" --filter="N111GQ"` will result in entries with FU111 or N111GQ

To process a filter with "AND": `--filter="FU111&&N111GQ"`

4.5. Scenario/Recording Utilities

4.5.1. ATCoach Conversion

The ATCSRVR utility listens for the STARS Maintenance LAN's multicast feeds from ATCoach and converts them to standardized formats. ATCSRVR can process radar, DASR, ADSB, MLAT, Interfacility, ETMS, and SIM data. SDRR configuration files are needed to prepare network devices and serial devices. These configuration files can take in values from a device file if there is one present. The utility can be executed entirely from the command line with the **atcsrv** executable. Or, a GUI can be launched with ATCSRVR.

```
> atcsrv
```

For the **ATCSRVR** GUI version, the default configuration file is **/usr/local/etc/atcsrv.0.conf**.

```
> ATCSRVR
```



Figure 42. ATCoach Conversion GUI

4.5.2. SDRR Player

The SDRR Player utility creates playlists containing SDRR scenarios. This utility can be launched from an icon on the KDE desktop (if created) or by entering ***sdrplayer*** (case sensitive) in a terminal window. This will launch the SDRR Player GUI interface:

```
> sdrplayer
```



Figure 43. SDRR Player Icon

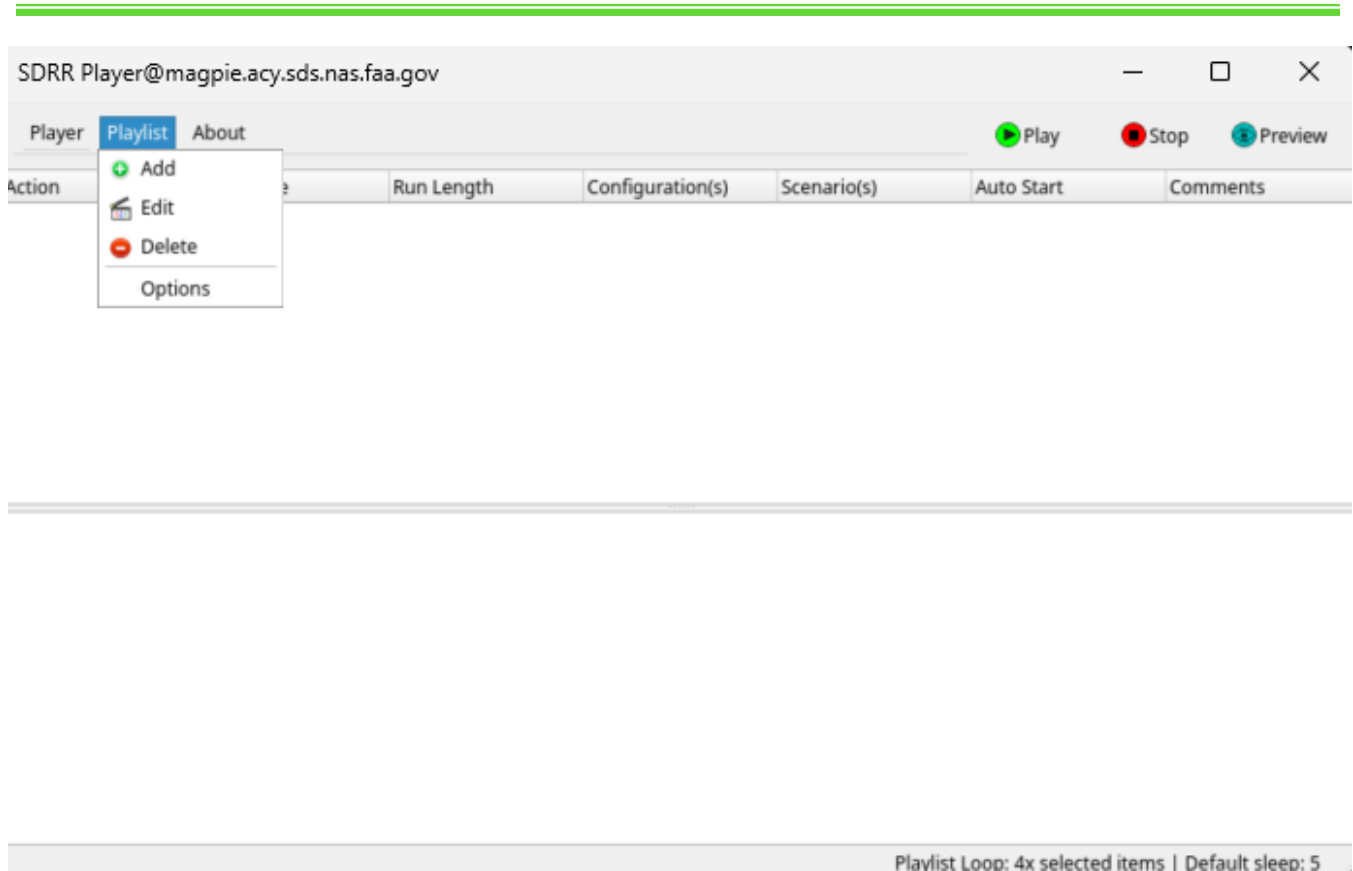
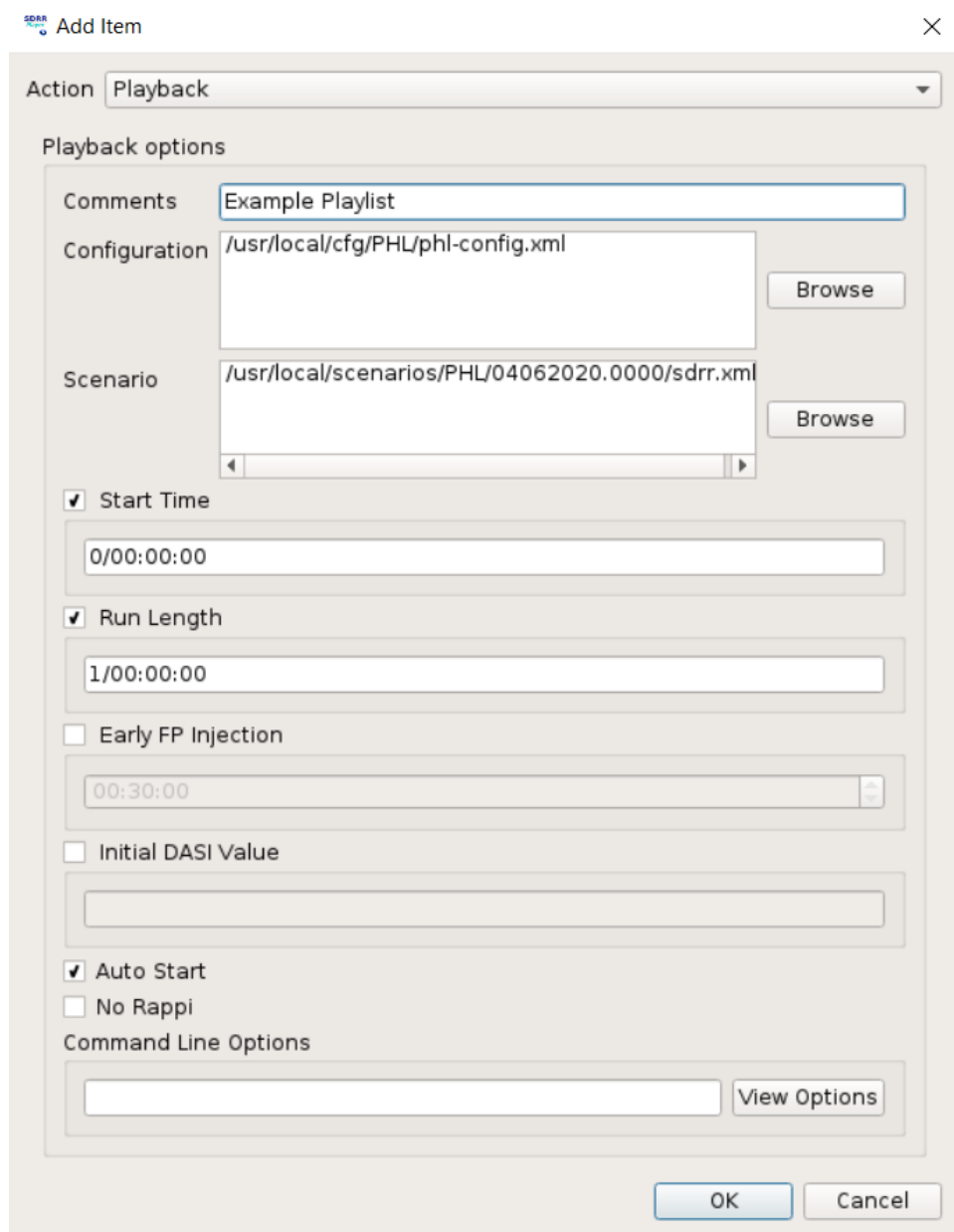


Figure 44. SDRR Player GUIAbout:

The About option displays the “About SDRRPLAYER” dialog which shows the version of SDRRPLAYER. To add a scenario to the playlist, users can select Playlist then select Add. The Add Item window will be displayed:



SDARS Add Item

Action: Playback

Playback options

Comments: Example Playlist

Configuration: /usr/local/cfg/PHL/phl-config.xml Browse

Scenario: /usr/local/scenarios/PHL/04062020.0000/sdrr.xml Browse

☒ Start Time
0/00:00:00

☒ Run Length
1/00:00:00

☐ Early FP Injection
00:30:00

☐ Initial DASI Value

☒ Auto Start

☐ No Rappi

Command Line Options
View Options

OK Cancel

Figure 45. Add Item Window

Action

Selection of “Playback” or “Sleep” to add either a scenario or a delay to the playlist.

Comments

Comments may be added for each item in the playlist.

Configuration

The full path may be entered or the “Browse” button may be pressed to select the configuration file(s).

Scenario

The full path may be entered or the “Browse” button may be pressed to select the scenario file(s).

Start Time

This field will be automatically populated when the scenario is selected, however users can adjust the start time.

Run Length

This field will be automatically populated when the scenario is selected, however users can adjust the run length.

Early FP injection

When the box is checked, users can set the amount of time prior to the “Start Time” to begin injecting FP messages.

Initial DASI value

When the box is checked, users can set the DASI value before the start of the scenario.

Auto Start

Checked by default; the scenario will automatically start.

No Rappi

Unchecked by default; when checked the scenario will run without a Rappi tab.

Command Line Options

Allows users to enter additional SDRR command line entries. Clicking on the “View Options” button displays the complete list. The device file name can be called out here while using a DMS report file configuration file.

```
--deviceFile=/usr/local/etc/[deviceFile]
```

OK

Once OK is clicked, the item is added to the playlist.

Items in the playlist window can be dragged-and-dropped to change the order. If the CTRL key is pressed while dragging-and-dropping the item, a copy will be created and added to the playlist. A plus sign with a box around it will be displayed to indicate the copy action.

Once all the items have been added to the playlist, the “Play” button can be pressed to begin running through the playlist. The playlist will play once unless set to replay. To set a replay, users can select **Playlist** and then select **Options** to set the amount of times for the playlist to loop.

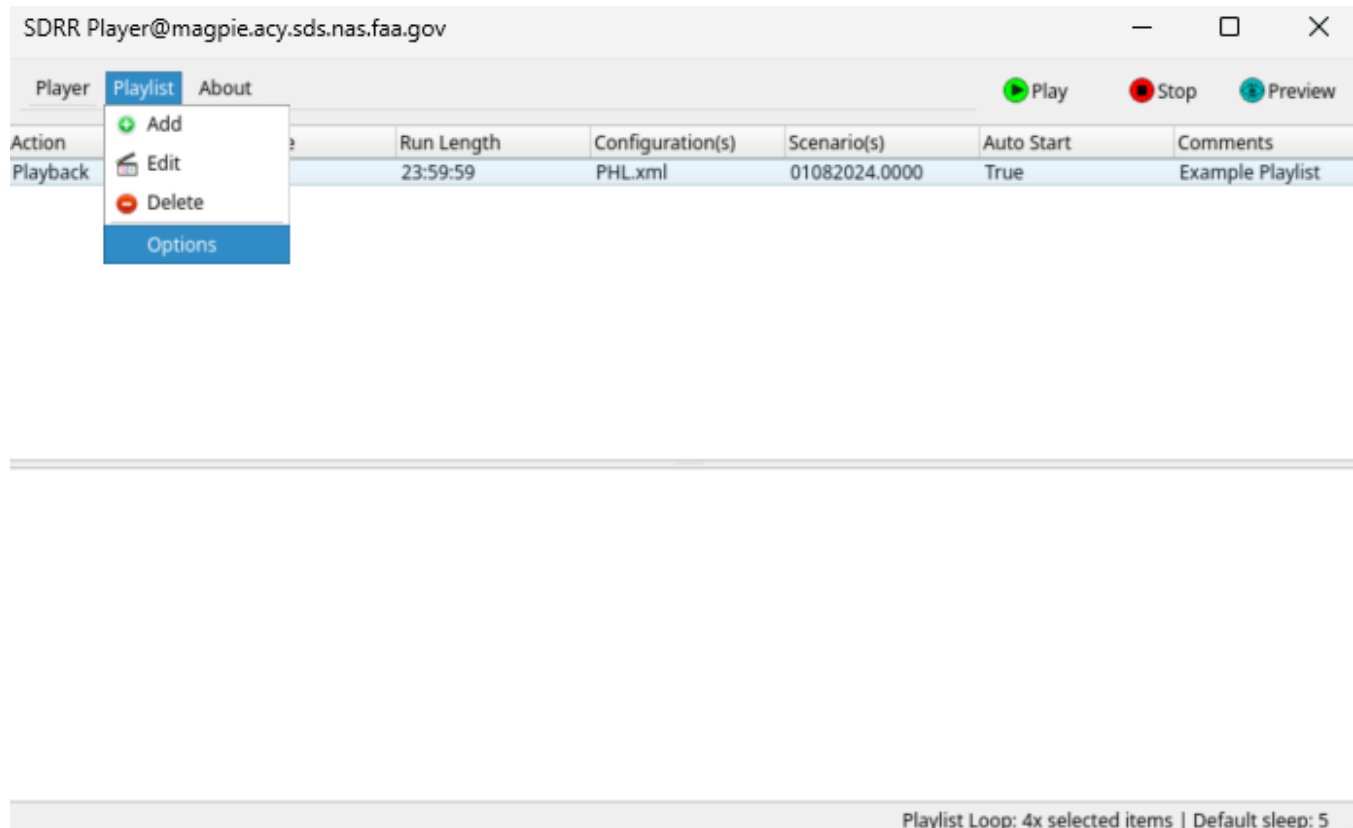


Figure 46. Playlist Options

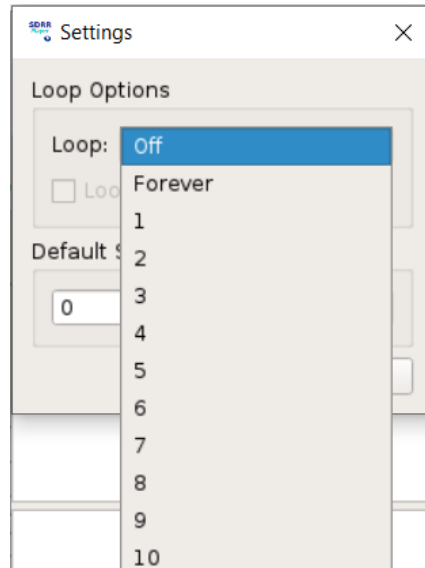


Figure 47. Loop Options

The playlist can be saved by selecting **Player** and then selecting **Save As**:

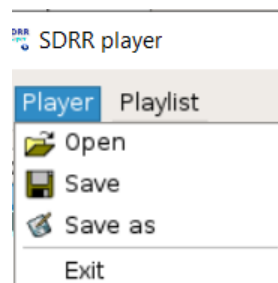


Figure 48. Saving the Playlist

A “Save Current Playlist” dialog will be displayed.

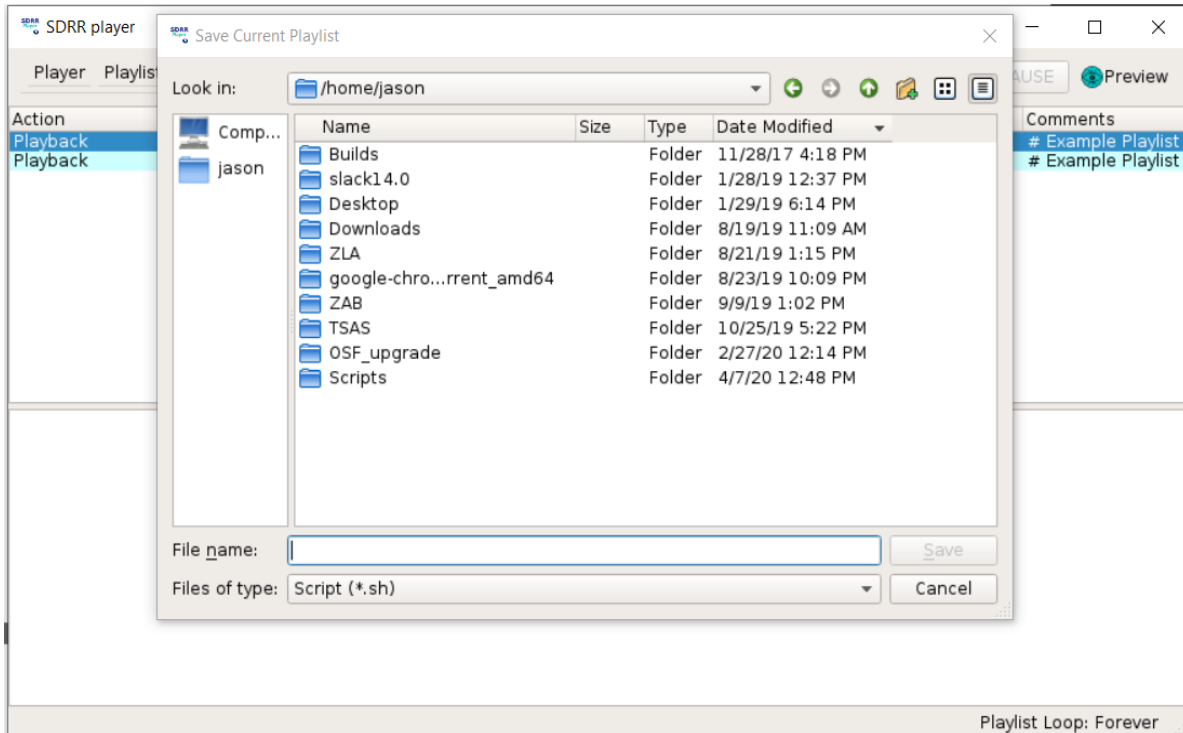


Figure 49. Save Current Playlist Dialog

Once the SDRR scenario opens, the SDRR Player should be used to stop and restart the scenario.

4.5.3. TARP

The **Time-based Archive Recording Player (TARP)** utility is a graphical interface that allows the user to either preview or playback an SDRR recording. This utility can be launched either from an icon on the Desktop (if created) or by typing **TARP** (case sensitive) in a terminal window. This will launch the TARP GUI interface:



Figure 50. TARP Icon

> TARP

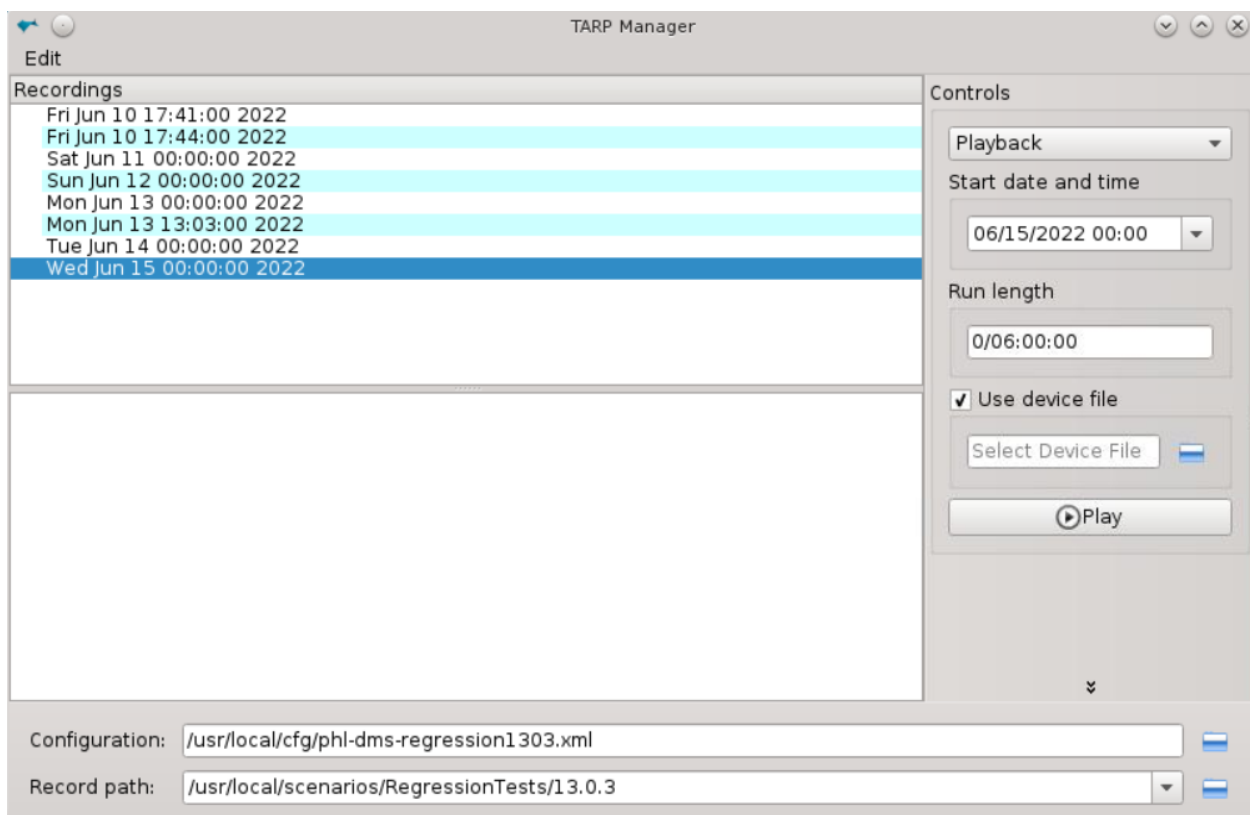


Figure 51. TARP Utility

The desired configuration can be selected under the “Configuration” section. The specified configuration can be used for both preview and playback. TARP will dynamically determine the devices needed depending on playback or preview mode. The “Record path” defaults to the directory specified by the environment variable `${RECORD_PATH}`, where TARP recordings are typically stored. The user can select “Playback/Preview” from the drop down box under the “Controls” section. When preview is selected the user will have the ability to specify an optional display file for the RAPPI. Recordings are selectable in the Recordings list view or by filling in a desired date and time in the “Date/Time” field under the “Controls” section. “Run length” is selectable and controls the playback’s runtime length.

The user can also set other optional settings under the “Edit” menu. When selected, the settings dialog will be displayed

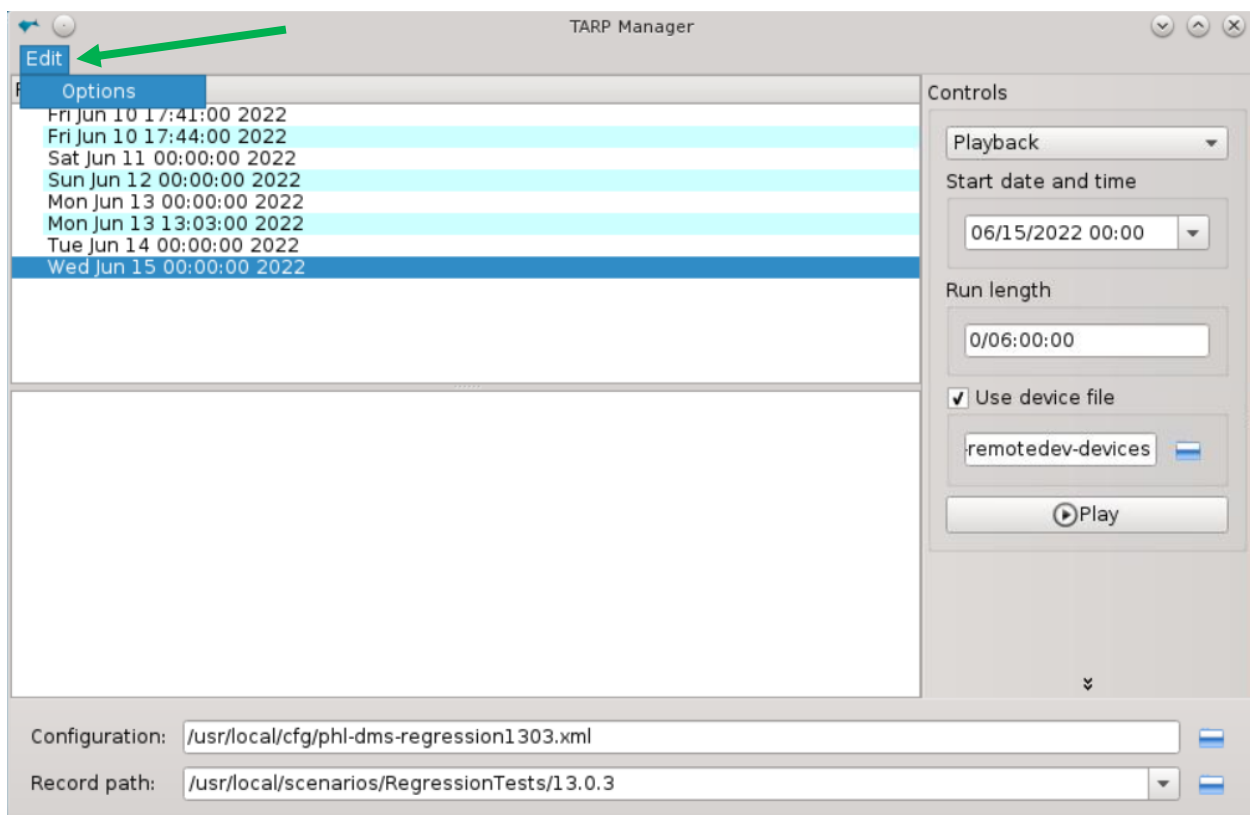


Figure 52. TARP Option

Destination

Where local and remote scenarios are stored for playback/preview.

Proxy Server

Proxy servers and Ports are specified for remote access from within a firewall.

Proxy Port

Proxy servers and Ports are specified for remote access from within a firewall.

Run Length

If selected the scenario will only execute for the specified length of time.

Auto Start

Start the Playback/Preview immediately.

Preview in real time

By default preview processes the data at high speed. Select this option to preview in real time.

Convert Facility names

Handles Center ID conversion found in the config.xml file, may be needed for CDR playback data.

Disable connection to psync daemon

Disables the psync timer required for En Route ADS-B.

UTC mode

Enables scenario IFDT and IP radar UTC time adjustment to current UTC time.

TARP allows remote recordings to be played locally by selecting the “Mount remote file system” button, shown below. Once selected, the “User”, “Host”, and “Path” should be entered of the remote SDRR machine to access. Once entered, this information can be saved for future use by selecting the “Save” button and providing a name. The “Host” can be an IP address or the hostname of the remote machine defined in the /etc/hosts file. The “Path” defaults to the directory specified by the environment variable `${RECORD_PATH}`, where most TARP recordings reside, but can be changed if needed. The “Device File” option must be used with any DMS report file configuration. Click Device File and select the appropriate device for that machine.

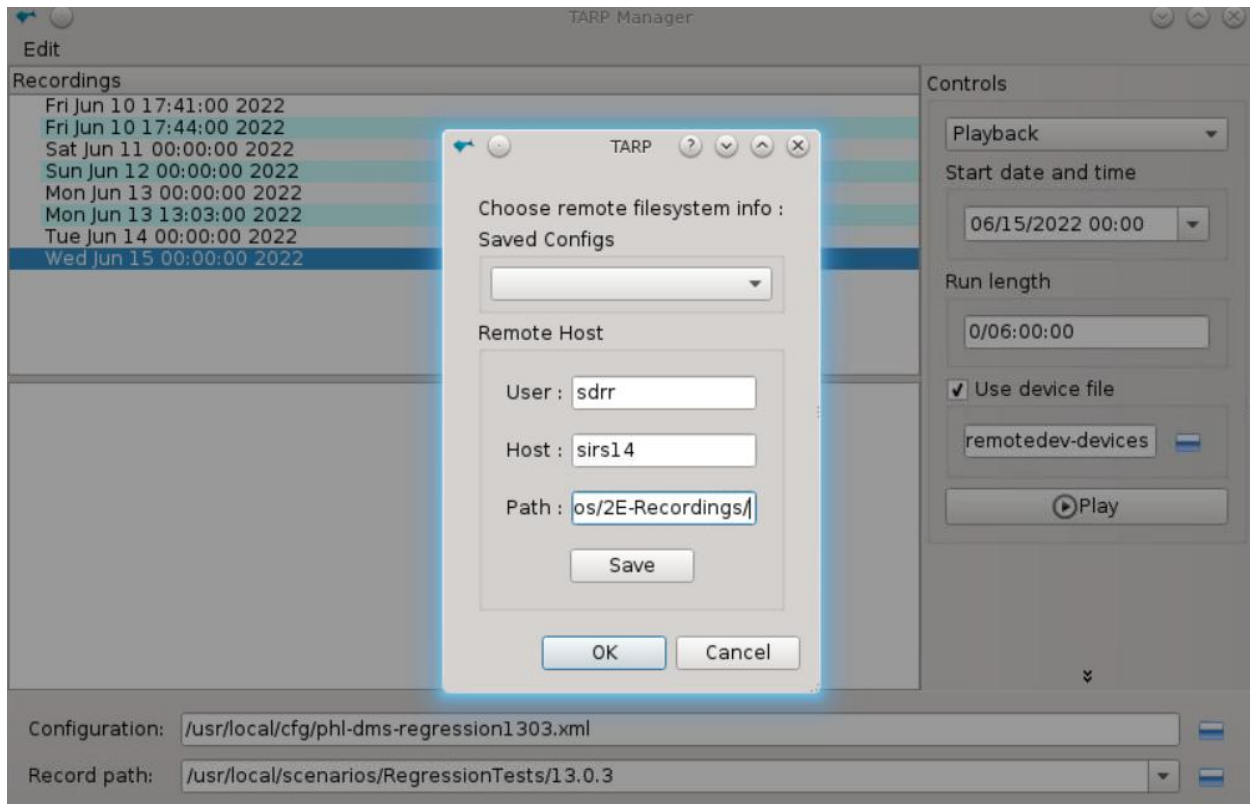


Figure 53. Remote Host

Once the Remote Host information is entered, selecting the “OK” button connects to a remote file location.

For remote access, the remote password will need to be entered once prompted. Enter the password for the given “User” and select “OK”.

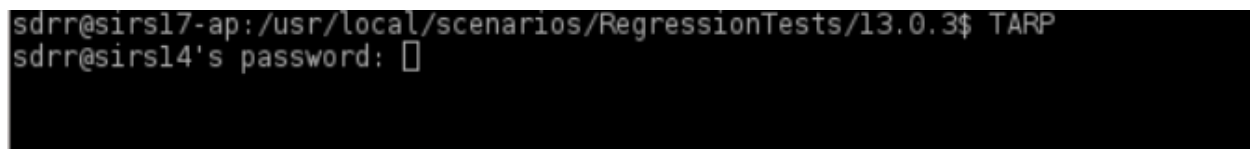


Figure 54. Remote Password

NOTE: Password prompt may appear in a pop-up dialog box in earlier versions.

The remote archive can now be selected for playback. The initial start may take longer than normal to load, as first time recordings are copied to the local machine. Notice below the location of the remote machine and user is displayed under the “Mount remote filesystem” button.

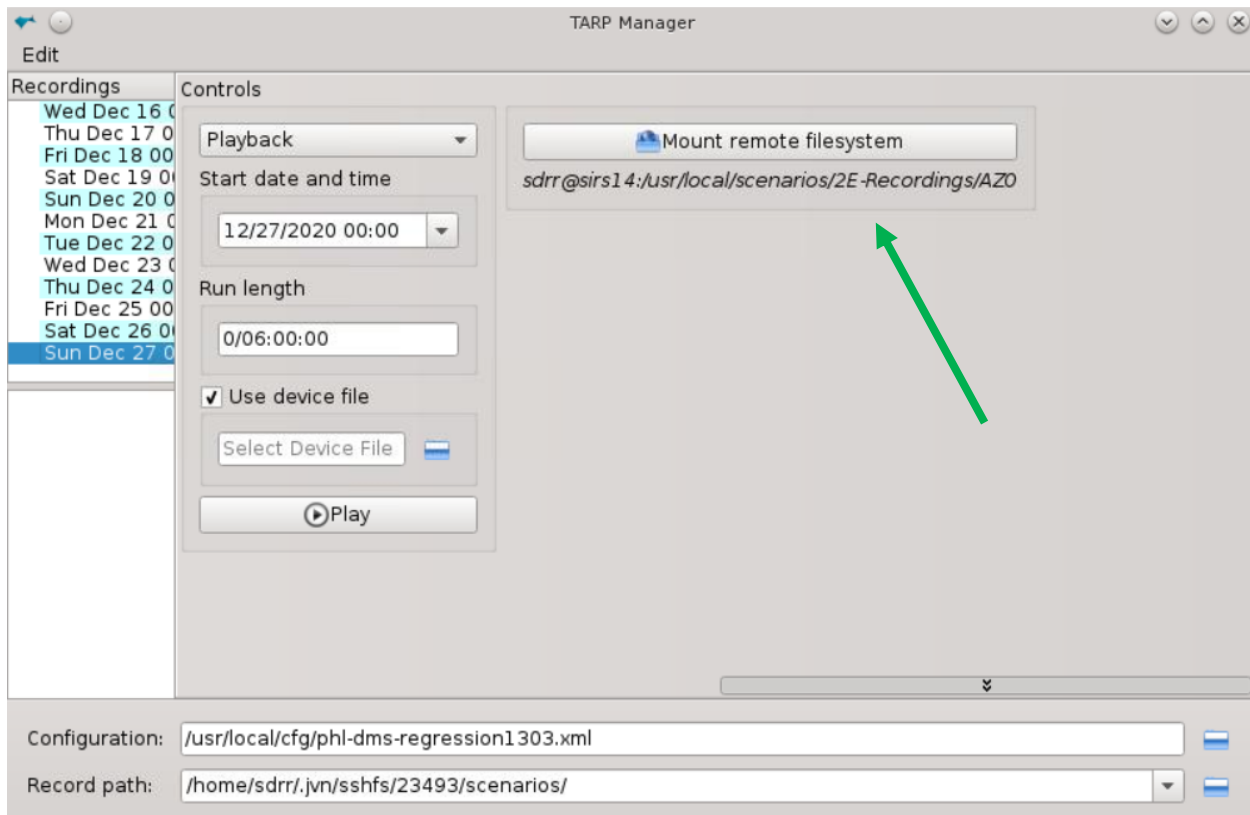


Figure 55. Remote Host and User

Record path

Populates with the temporary storage location of the mounted recording.

Play

Select the Play button once all Control options and recordings have been selected.

SDRR should start in a normal fashion with options selected. TARP will remain open in the background and provide log based view of the playbacks details, including errors and new file storage locations.

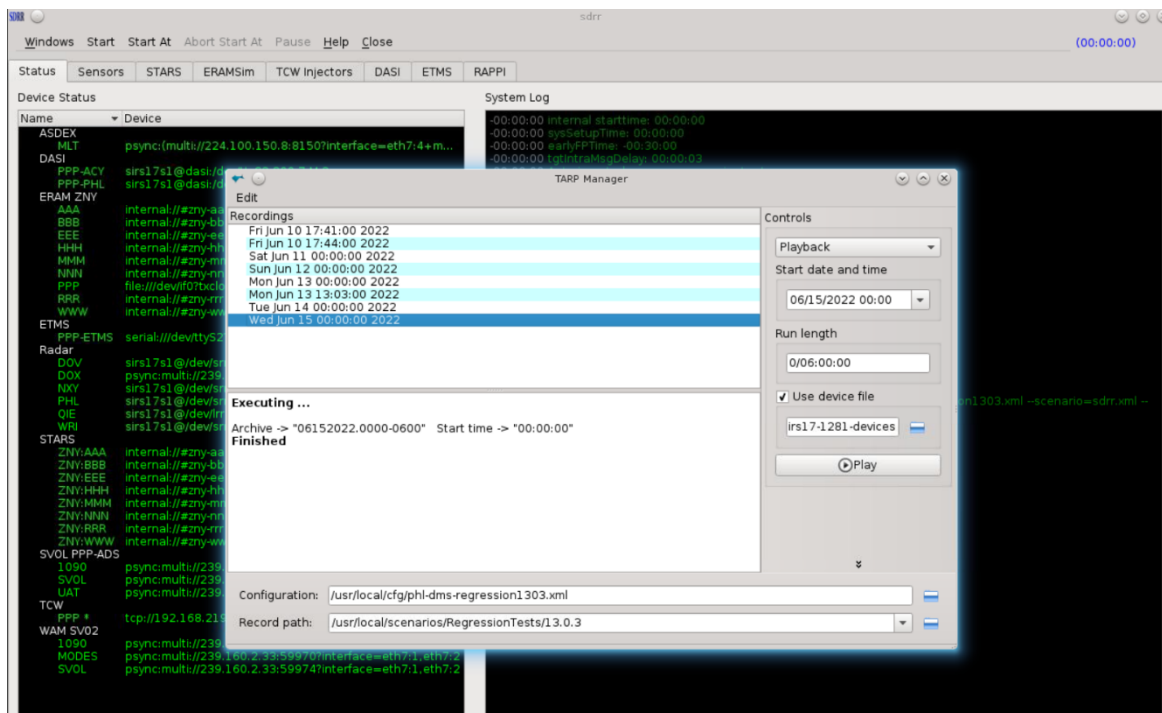


Figure 56. SDRR is Launched



Figure 57. Playback Details

TARP can be used to quickly preview recordings by invoking the AViD viewer by selecting “Preview” under the “Controls” panel.

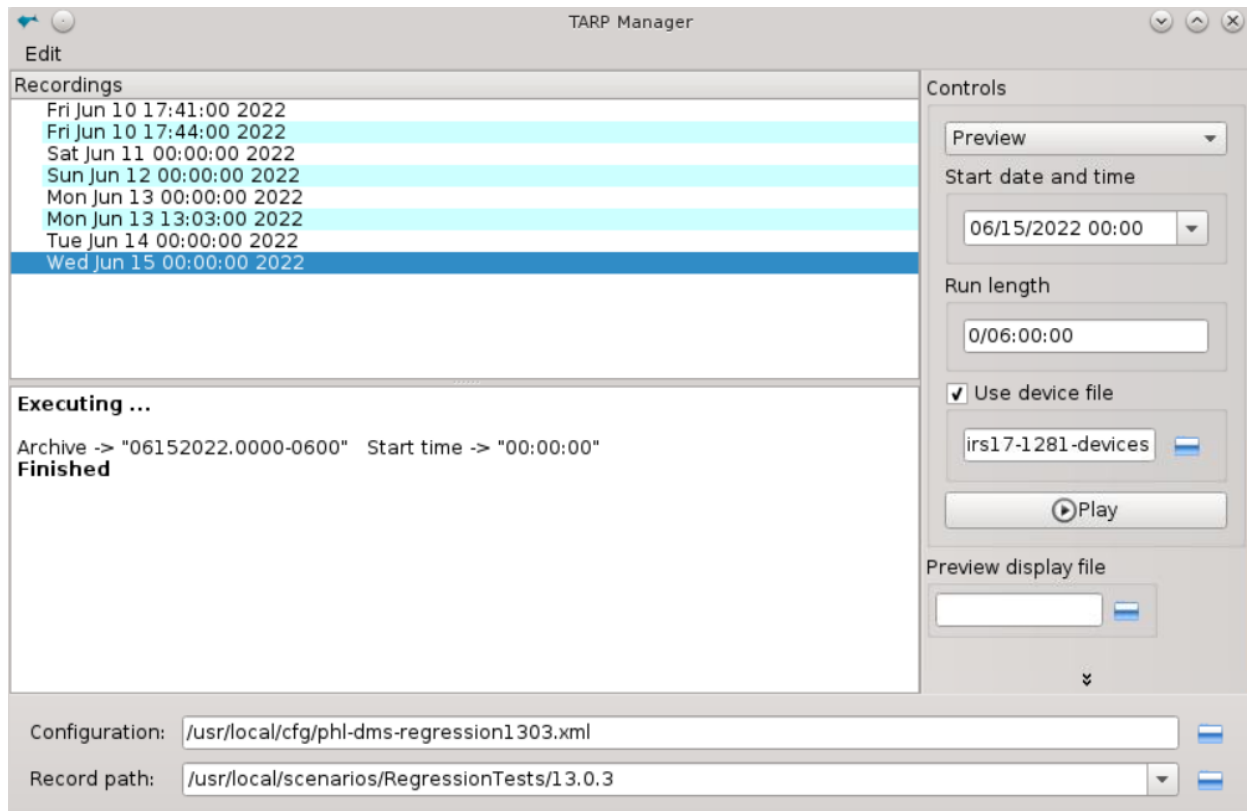


Figure 58. TARP Preview

Once Preview is selected, the "Preview display file" dialog is available and users can select a custom map based display.xml. If nothing is selected the default US map is loaded under the config.xml file attributes. Select "Play" to open the Preview mode AViD viewer.

Once AViD opens the users can Run, Step and Pause the preview. The Run button is 60x faster than standard or real-time playback. The Step button allows for scan by scan updates per click and is a benefit when detailed analysis is required. AViD functions are available and are needed to display targets. Users will need to Select the View tab and select the desired Data/Sensor. For more information on AViD functions, please see the AViD user manual.

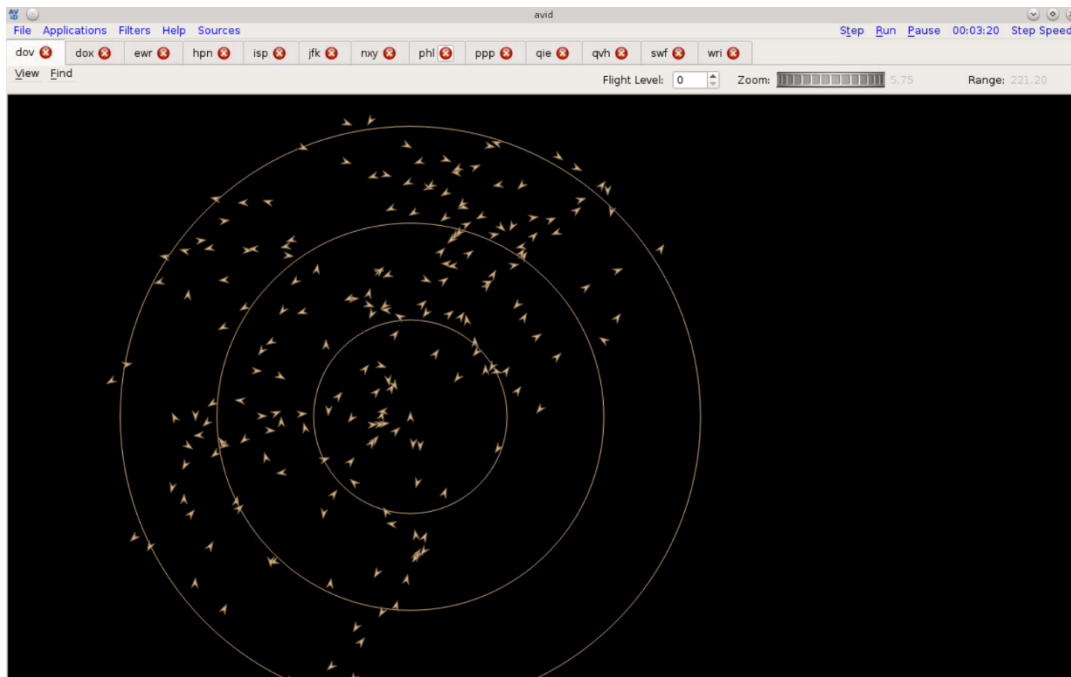


Figure 59. AVID GUI

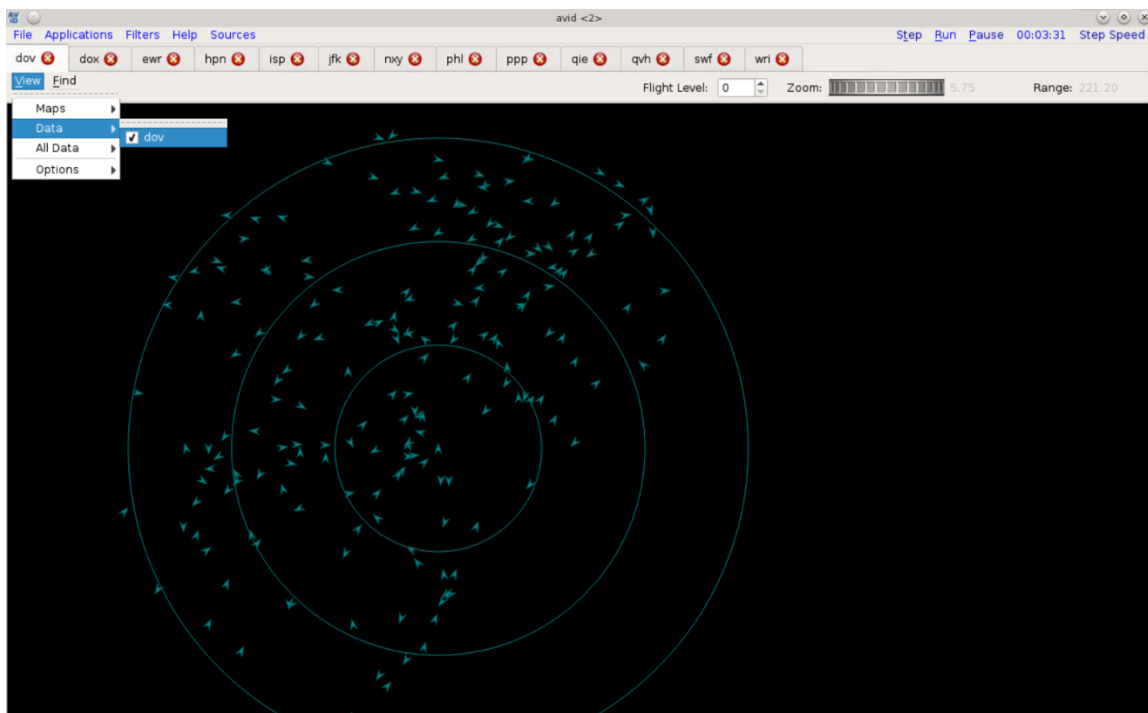


Figure 60. AVID Radar Filter

4.5.4. Scenario Export Utility – exportsdrr

The **exportsdrr** utility is used to generate a static SDRR scenario via the command line. The utility accepts the following parameters as inputs.

Table 4. exportsdrr Parameters

Parameter	Description
Modes *note: atleast one mode must be selected	
--nonSurveillance	
--surveillance	
Surveillance Options	
--nowx	don't generate radar wx
--hgiEmulation	perform cv4400 conversion where needed
--nonRadarAreas="file"	file containing xml definition of non radar areas
--trackUpdateInterval=seconds	if not specified, defaults to 30. (for tracks.xml)
--rangeBias=ft	set range bias for radars
--azBias=deg	set azimuth bias for radars
--rangeDev=ft	set range noise std deviation for radars
--azDev=deg	set azimuth noise std deviation for radars
--ignoreHorizonTest	sets flag to ignore horizon test
--noStaticMsgs	
NonSurveillance options	
--quiet	
--rsiFile="file"	get RSIs from specified ATCoach control file.
Miscellaneous	
--terminalOnly	export for terminal only
--ignoreRsis	ignore specified rsis (separated by comma)

Parameter	Description
--runTime	default is 06:00:00
--aid=aid	filter by aid
--datacomm	Special export for datacomm testing
--tfdm	Special export for TFDM testing
--tcwMsgs	Include TCW handoff msgs in export

The syntax for **exportsdrr** is as follows:

```
> exportsdrr "dir" {mode(s)} [options]
```

In practice the command will exclude the "" around dir (directory), the {} around modes, and the [] around options.

The following is an example of an exportsdrr command that creates a scenario called phl-test with surveillance on and a running time of 1 hour:

```
> exportsdrr phl-test --surveillance --runTime 01:00:00
```

```
sdrr@dogfish:/usr/local/gsgt/Regression1263$ exportsdrr phl-test --surveillance --runTime 01:00:00
Opening scenario...
Parsing airspace adaptation...
Parsing fac adaptation...
Importing preferences from prefs.xml
Finished Preferences
Importing RSIs from rsi.xml
1 RSIs Imported
Imported 1 wx slices
Importing system commands from syscmds.xml
0 System Commands Imported
Importing targets from tgts.xml
Target MS01 contains non ICAO departure airport PHL
Target MS01 contains non ICAO arrival airport EWR
Target ADSB01 contains non ICAO departure airport PHL
Target ADSB01 contains non ICAO arrival airport ACY
Target ARSA01 contains non ICAO departure airport PHL
Target ARSA01 contains non ICAO arrival airport WRI
22 Targets Loaded from tgts.xml
Starting export
Creating directories
Constructing scenario msglist
Generating surveillance data...
Creating radar messages
Exporting nexrad...
sdrr@dogfish:/usr/local/gsgt/Regression1263$ ls
Regression1263_210326.1227/  exportsdrr.html  phl-test/  rdrtgcnt.html  rsi.xml  sortedlisting.html  trklist.ARR04.html
activeRunways.xml          favlist.html     pilotControls.xml  rdrtgcnt.png  scenario.xml  syscmds.xml       trklist.html
devctl.xml                 gsgt.cfg        prefs.xml          restrictions.xml  scriptDefinitions.xml  tgts.xml
sdrr@dogfish:/usr/local/gsgt/Regression1263$
```

Figure 41. Example of exportsdrr

4.5.5. Special Message Injection Utility

A special message injection utility called `jmsTest` allows users to inject an RTCS message with an invalid format. The normal SDRR message injection function ensures that RTCS messages have correctly formatted properties and XML structure. The `jmsTest` utility allows a message to be injected where the properties or XML structure are malformed.

To execute the `jmsTest` utility, enter:

```
> jmsTest "devName" [--file="fileName"] (--writeOnly | --readOnly)
```

For example:

```
> /usr/local/jvn.13.7.6/bin/jmsTest "solace://tbfmsolace01.tbfm.leidos.com:55003?
queueName=RTCPublish_08&username=solace&passwd=solace1&compressed&vpn=TBFM_SW_IS_
DEPLOYMENT --file=jmsMsg.xml --w
```

The malformed messages are scripted in the example `jmsMsg.xml` file:

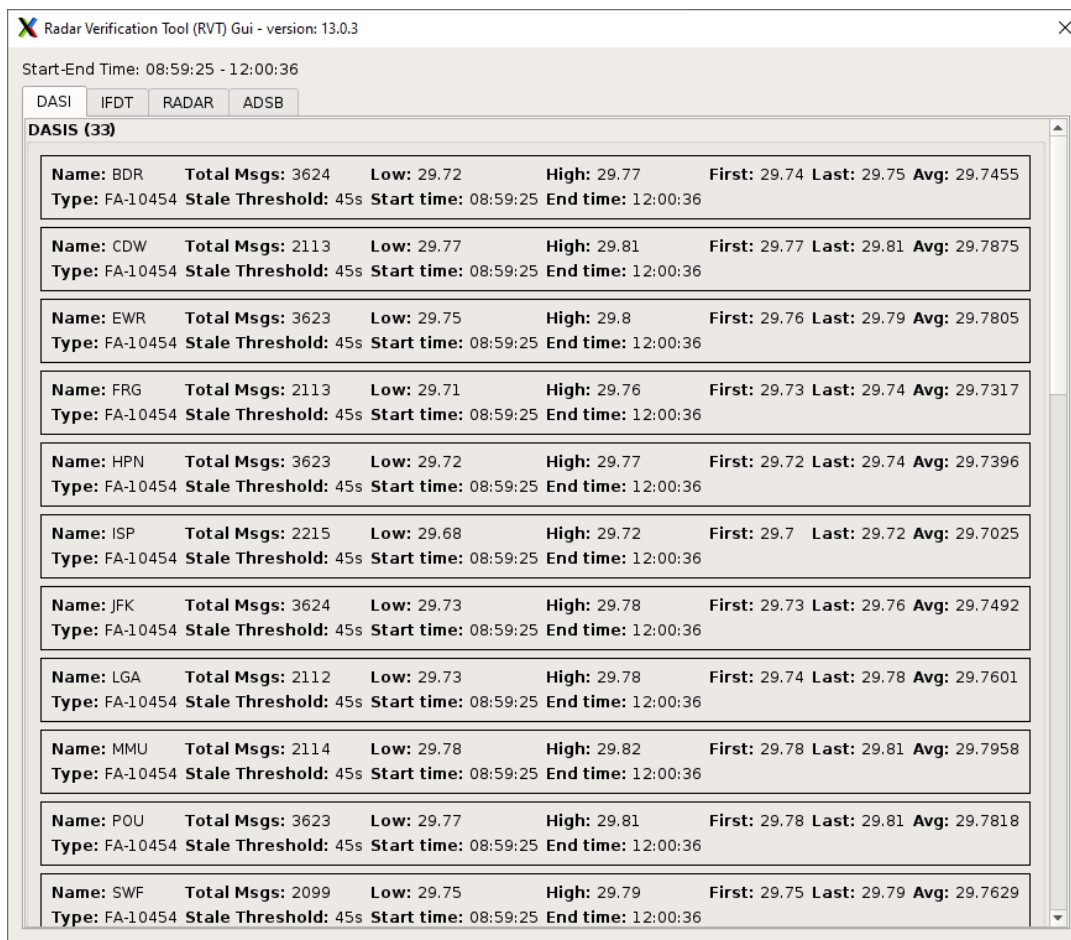
```
DPT:KLAX
MSG_NAME:rtcsHb
REQ_ID:ALL
VER:2.0.0
<rtcsMsg injectorName="klax" time="-00:05:00.00">
  <rtcs xmlns="urn:us:gov:dot:faa:atm:tfm:releasetimecoordination:1.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:us:gov:dot:faa:atm:tfm:releasetimecoordination:1.0.0 RTCS-
jmsdd.xsd">
  <relReq>
    <acid>%AID%</acid>
    <gufi>%GUFi%</gufi>
    <tmaId>%TMAID%</tmaId>
    <schAct>HEDULE</schAct>
    <reime>%EPOCH/00:06:05%</reqTime>
    <rw>24R</rw>
  </relReq>
</rtcs>
</rtcsMsg>
```

4.5.6. Radar Verification Tool (RVT)

Radar Verification Tool (RVT) is a GUI and command line utility used for verifying scenario data prior to SDRR playback. RVT uses STARS adaptation files (.rpt) to compare present scenario data. RVT looks for things like, missing SVOL, Radars, Radar Channels, DASI and IFDT messages.

To execute the **rvt** utility, enter:

```
> /usr/local/jvn.13.7.6/bin/rvt --dmsFile=FILE --scenDir=DIRPATH [-c]
--scenDir DIRPATH and dmsFile FILE should be absolute paths
-c = commandline (no GUI)
```



Radar Verification Tool (RVT) Gui - version: 13.0.3

Start-End Time: 08:59:25 - 12:00:36

DASI IFDT RADAR ADSB

DASI (33)

Name: BDR	Total Msgs: 3624	Low: 29.72	High: 29.77	First: 29.74	Last: 29.75	Avg: 29.7455
Type: FA-10454 Stale Threshold: 45s Start time: 08:59:25 End time: 12:00:36						
Name: CDW	Total Msgs: 2113	Low: 29.77	High: 29.81	First: 29.77	Last: 29.81	Avg: 29.7875
Type: FA-10454 Stale Threshold: 45s Start time: 08:59:25 End time: 12:00:36						
Name: EWR	Total Msgs: 3623	Low: 29.75	High: 29.8	First: 29.76	Last: 29.79	Avg: 29.7805
Type: FA-10454 Stale Threshold: 45s Start time: 08:59:25 End time: 12:00:36						
Name: FRG	Total Msgs: 2113	Low: 29.71	High: 29.76	First: 29.73	Last: 29.74	Avg: 29.7317
Type: FA-10454 Stale Threshold: 45s Start time: 08:59:25 End time: 12:00:36						
Name: HPN	Total Msgs: 3623	Low: 29.72	High: 29.77	First: 29.72	Last: 29.74	Avg: 29.7396
Type: FA-10454 Stale Threshold: 45s Start time: 08:59:25 End time: 12:00:36						
Name: ISP	Total Msgs: 2215	Low: 29.68	High: 29.72	First: 29.7	Last: 29.72	Avg: 29.7025
Type: FA-10454 Stale Threshold: 45s Start time: 08:59:25 End time: 12:00:36						
Name: JFK	Total Msgs: 3624	Low: 29.73	High: 29.78	First: 29.73	Last: 29.76	Avg: 29.7492
Type: FA-10454 Stale Threshold: 45s Start time: 08:59:25 End time: 12:00:36						
Name: LGA	Total Msgs: 2112	Low: 29.73	High: 29.78	First: 29.74	Last: 29.78	Avg: 29.7601
Type: FA-10454 Stale Threshold: 45s Start time: 08:59:25 End time: 12:00:36						
Name: MMU	Total Msgs: 2114	Low: 29.78	High: 29.82	First: 29.78	Last: 29.81	Avg: 29.7958
Type: FA-10454 Stale Threshold: 45s Start time: 08:59:25 End time: 12:00:36						
Name: POU	Total Msgs: 3623	Low: 29.77	High: 29.81	First: 29.78	Last: 29.81	Avg: 29.7818
Type: FA-10454 Stale Threshold: 45s Start time: 08:59:25 End time: 12:00:36						
Name: SWF	Total Msgs: 2099	Low: 29.75	High: 29.79	First: 29.75	Last: 29.79	Avg: 29.7629
Type: FA-10454 Stale Threshold: 45s Start time: 08:59:25 End time: 12:00:36						

Figure 56. RVT DASI

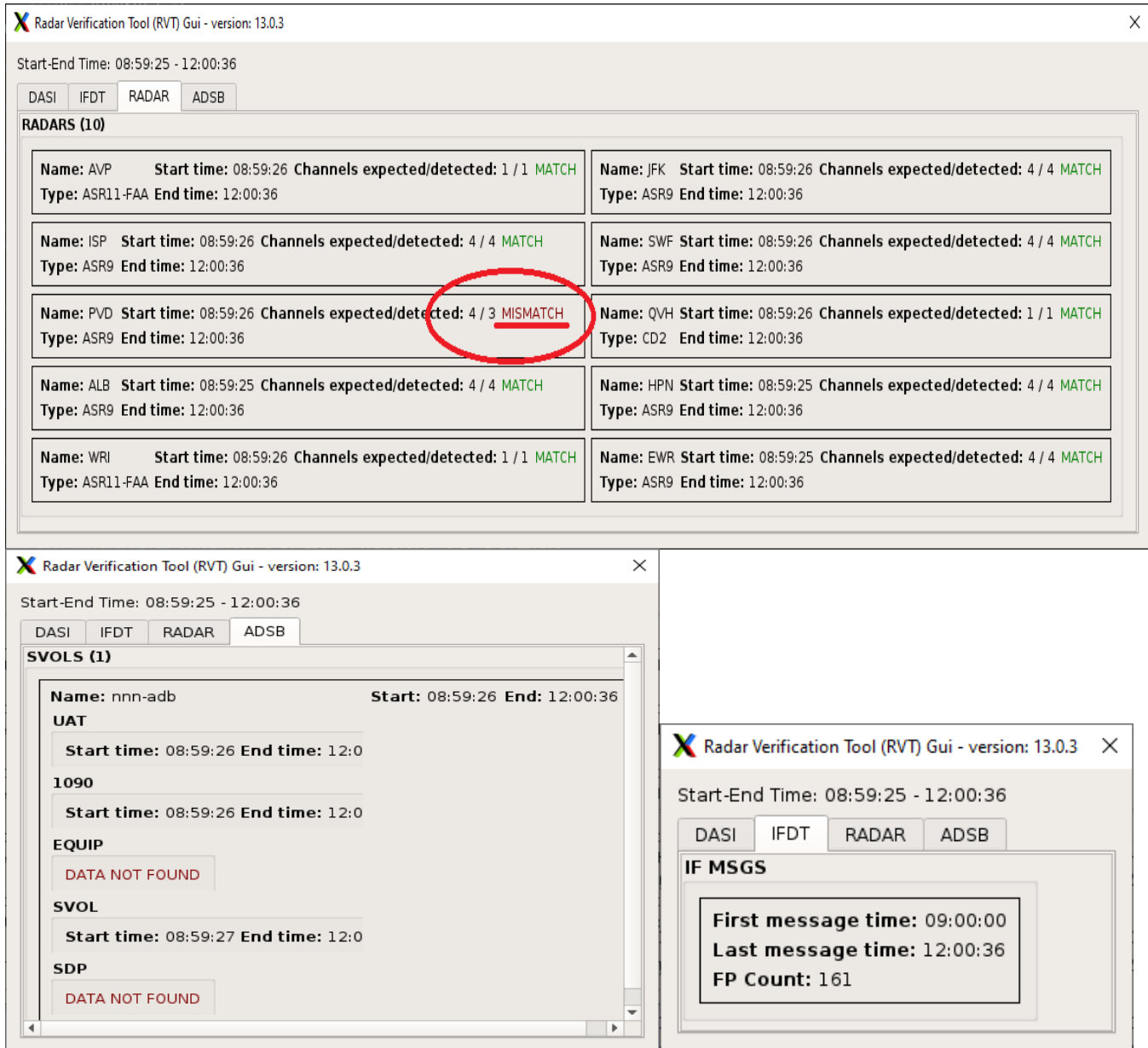


Figure 57. RVT Radar & IFDT

Appendix A. Acronyms

ACID	Aircraft Identification
ADS-B	Automatic Dependent Surveillance – Broadcast
ARSR	Air Route Surveillance Radar
ARTCC	Air Route Traffic Control Center
ARTS	Automated Radar Terminal System
ASR	Airport Surveillance Radar
ASR-9	Airport Surveillance Radar Model-9
ASTERIX	All Purpose Structured Eurocontrol Radar Information Exchange
ATC	Air Traffic Control
AviD	Airspace Visualization Display
BRTQC	Beacon Real Time Quality Control
CAS	Commercially Available Software
CD	Common Digitizer
CDR	Continuous Data Recording
CMS	Common Message Set
DASI	Digital Altimeter Setting Indication System
DASR	Digital Airport Surveillance Radar
DYSIM	Dynamic Simulation
ECG	External Communications Gateway
ECGP	External Communications Gateway Protocol
EDDS	En Route Data Distribution System
ERAM	En Route Automation Modernization
ETMS	Enhanced Traffic Management System
FAA	Federal Aviation Administration
FDIO	Flight Data Input/Output
GSGT	Graphic Simulation Generation Tool

GUI	Graphical User Interface
IFDT	Interfacility Flight Data Transfer
Mode 3/A	Identification Reporting Mode of Secondary Radar
Mode C	Altitude Reporting Mode of Secondary Radar
Mode S	Mode Select Beacon System
MLAT	Multilateration
NAS	National Air Space
RAPPI	Random Access Plan Position Indicator
RSI	Record Select Indicator
RTQC	Real Time Quality Control
RVT	Radar Verification Tool
SAC	System Area Code
SDRR	Simulation Driver Radar Recorder
SIC	System Identity Code
SIRS	STARS Interfacility and Radar Simulator
SSRV	Simulation Services
STARS	Standard Terminal Automation Replacement System
SWIM	System-Wide Information Management
TARP	Time-based Archive Recording Player
TBFM	Time Based Flow Management
TSIM	TBFM Simulation
TRACON	Terminal Radar Approach Control
WAM	Wide Area Multilateration
WJHTC	William J. Hughes Technical Center