JVN
Communications, Inc.

Simulation Driver and Radar Recorder (SDRR)

Operation and Maintenance Manual

(SDRR – OMM – 010)

Apr 13, 2011

Volume 1: Operation and Maintenance Manual SDRR Version 2.5.11
Revision 10

JVN Communications Inc.
707 White Horse Pike, Suite B6
Absecon, NJ 08201
609-569-1477

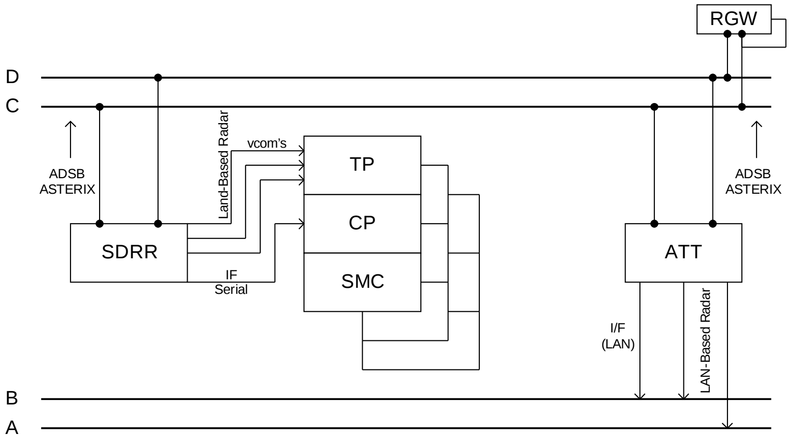
Table of Contents

1. SDRR Overview

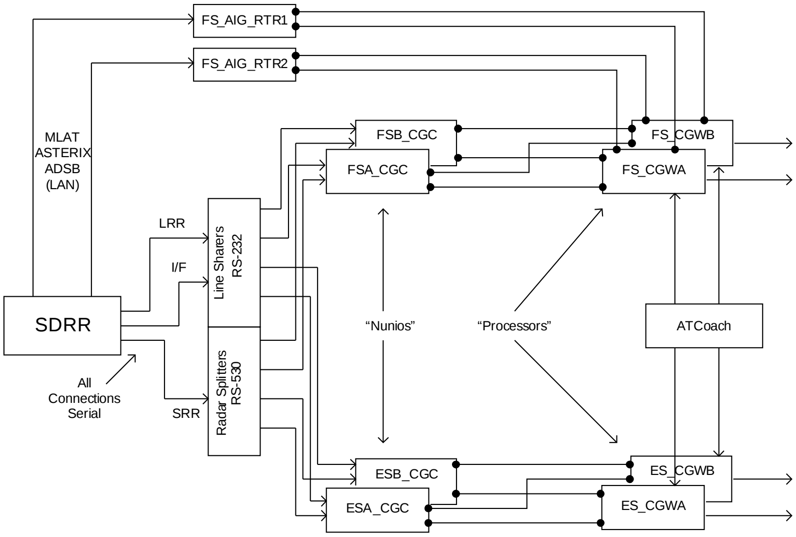
The Simulation Driver and Radar Recorder (SDRR) is a simulation injection tool that can be configured to run in various environments. It allows injection of local, interfacility and surveillance messages. It provides a display of all related NAS, ARTS and surveillance facilities. The display provides message counts, device status and error messages.

SDRR is an upgrade to the original suite of terminal simulation tools known as FIRS (FAA Interfacility & Radar Simulator). It incorporates familiar FIRS capabilities with the latest Linux tools available in the KDE desktop environment.

SDRR was designed by JVN Communications Inc. to provide surveillance data communications to terminal & enroute ATC systems. In addition, SDRR acts as a Host simulator communicating with terminal systems in a real-time environment. It can send flight plans and respond accordingly to all terminal interfacility messages. It can also play the existing or newly built scenarios into various terminal and host systems.



SDRR in ARTS

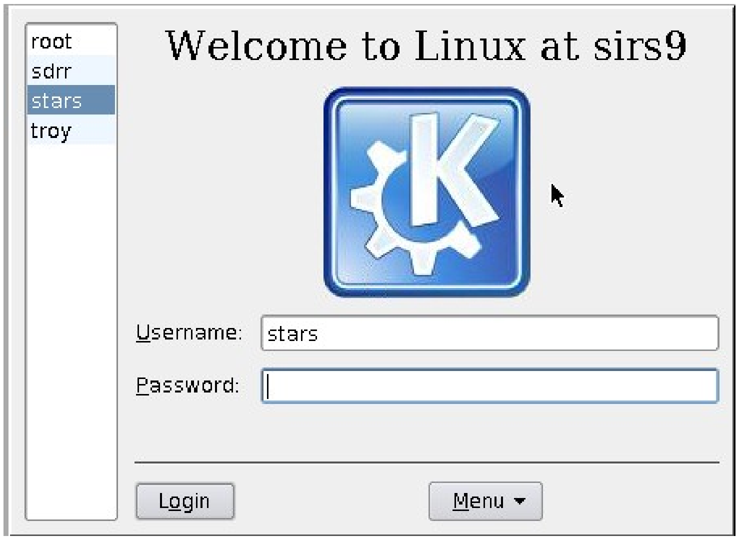


SDRR in STARS

2. Getting Started

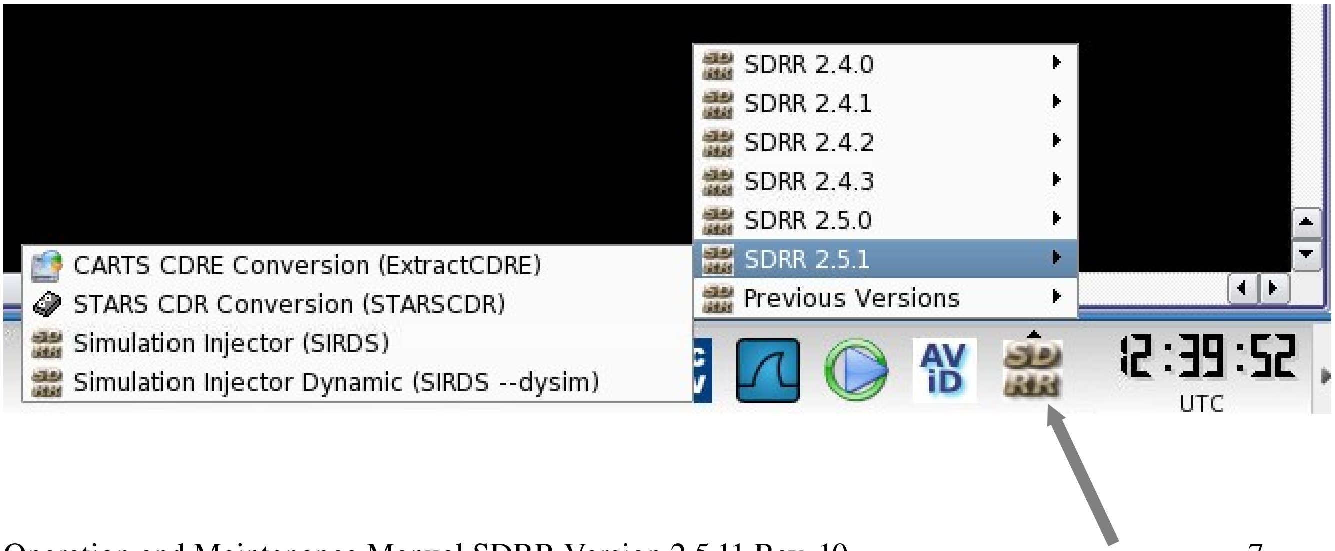
Your new SDRR tool is preconfigured to boot up to a login wizard which has all the users listed on the left side of the window. Simply select a user by left clicking on it and type in the password. Then click the login button at the bottom or just hit enter on the keyboard. After a short loading period the KDE desktop will appear. The Menu button includes options to Reboot or Halt the system.

NOTE: *Do not login as root if you intend to have access to the SDRR commands and utilities. They are disabled for the 'root' user by design. If you find that you need root access while logged in either log out and log in again as 'root' or just open a terminal window and type "su" and enter the 'root' password.*



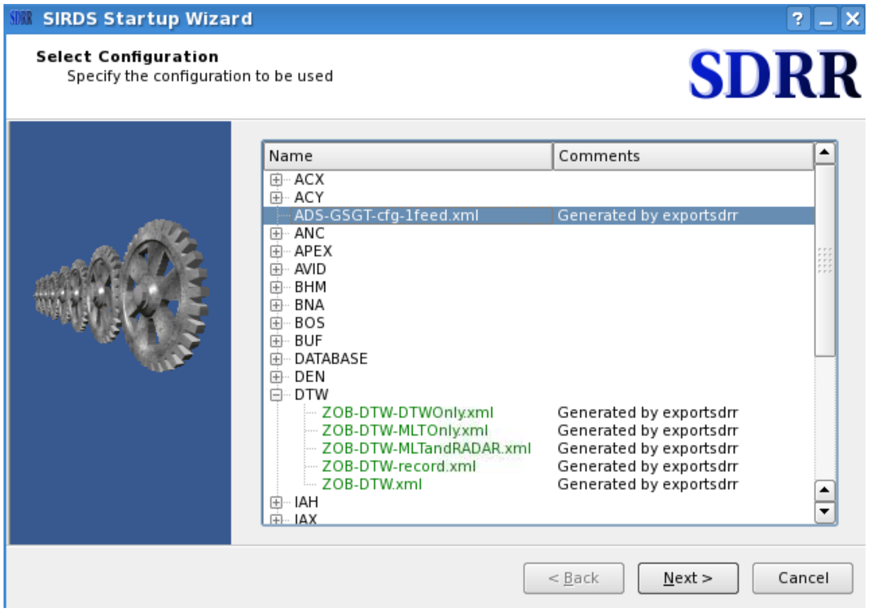
Typical KDE desktop

Once logged in, the SDRR can be started either by left click on the SDRR icon on the lower right side of the display (see illustration bellow) or by typing **sirds** at a command line. Notice below more than one version of SDRR is available. Once you select the desired version, you are given a few SDRR utilities as well as the SIRDS (Simulation Driver Radar Recorder) to choose from. The Dynamic SIRDS, the same as (sirds –dysim), is used in conjunction with GSGT for real-time playback manipulation. This is discussed later.


SDRR 2.4.0
SDRR 2.4.1
SDRR 2.4.2
SDRR 2.4.3
SDRR 2.5.0
SDRR 2.5.1
Previous Versions
CARTS CDRE Conversion (ExtractCDRE)
STARS CDR Conversion (STARSCDR)
Simulation Injector (SIRDS)
Simulation Injector Dynamic (SIRDS --dysim)
AV iD
SD RR
12:39:52
UTC

3. System Configuration

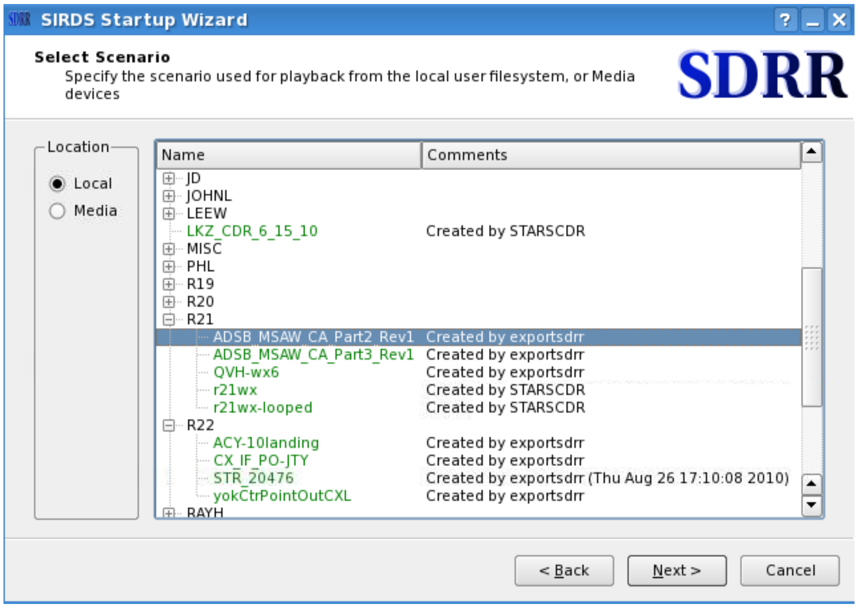
Once the SDRR is launched the Configuration menu appears. This shows expandable directories in **black** and selectable configuration files in **green**. For configuration files to appear selectable green, they need to end in .xml. Users may need to expand the **Name** bar to the right to see the entire filename. The **Comments** are displayed from the comments section of the configuration file. User can add or change these comments. These files are located in the **/usr/local/cfg** directory. Select the **Next** button to continue to scenario selection.



Select Configuration
Specify the configuration to be used

Name	Comments
⊞ ACX	
⊞ ACY	
ADS-GSGT-cfg-1feed.xml	Generated by exportsdrr
⊞ ANC	
⊞ APEX	
⊞ AVID	
⊞ BHM	
⊞ BNA	
⊞ BOS	
⊞ BUF	
⊞ DATABASE	
⊞ DEN	
⊟ DTW	
ZOB-DTW-DTWOnly.xml	Generated by exportsdrr
ZOB-DTW-MLTOnly.xml	Generated by exportsdrr
ZOB-DTW-MLTandRADAR.xml	Generated by exportsdrr
ZOB-DTW-record.xml	Generated by exportsdrr
ZOB-DTW.xml	Generated by exportsdrr
⊞ IAH	
⊞ IAX	

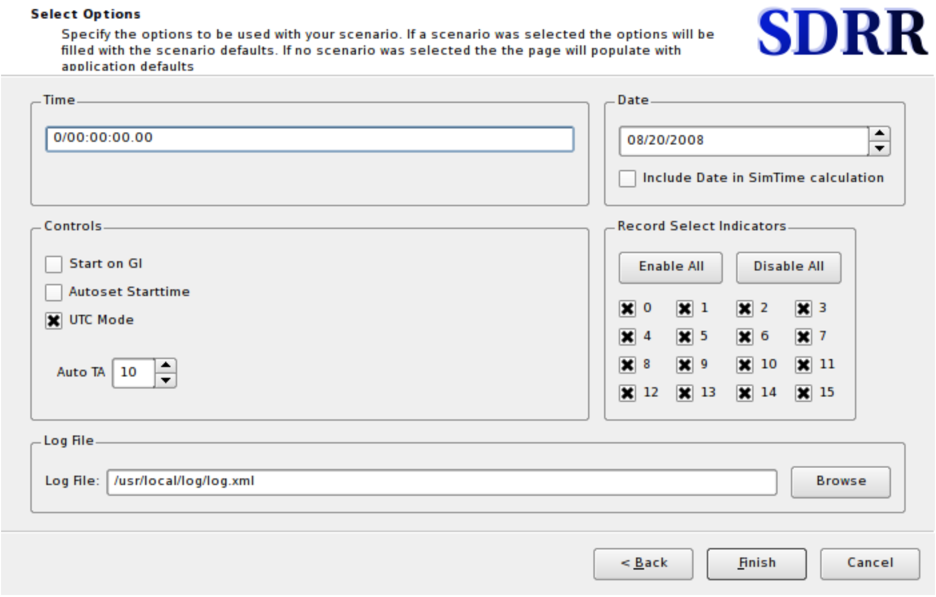
Select Scenario window displays **directories** and **scenarios**. This menu is populated from the **/usr/local/scenarios** directory. The example below demonstrates the R21 directory expanded and the ADSB_MSAW_CA_Part2_Rev1 scenario selected. Selecting a scenario is optional, you can click next without selecting a scenario to test your configuration file or test Interfacility connections. Click **Next** to continue.



Select Scenario
Specify the scenario used for playback from the local user filesystem, or Media devices

Location
● Local
○ Media

Name	Comments
⊞ JD	
⊞ JOHNL	
⊞ LEEW	
LKZ_CDR_6_15_10	Created by STARSCDR
⊞ MISC	
⊞ PHL	
⊞ R19	
⊞ R20	
⊟ R21	
ADSB_MSAW_CA_Part2_Rev1	Created by exportsdrr
ADSB_MSAW_CA_Part3_Rev1	Created by exportsdrr
QVH-wx6	Created by exportsdrr
r21wx	Created by STARSCDR
r21wx-looped	Created by STARSCDR
⊟ R22	
ACY-10landing	Created by exportsdrr
CX_IF_PO-JTY	Created by exportsdrr
STR_20476	Created by exportsdrr (Thu Aug 26 17:10:08 2010)
yokCtrPointOutCXL	Created by exportsdrr
⊞ RAYH	



Select Options

Specify the options to be used with your scenario. If a scenario was selected the options will be filled with the scenario defaults. If no scenario was selected the the page will populate with application defaults

Time: 0/00:00:00.00

Date: 08/20/2008

☐ Include Date in SimTime calculation

Controls

☐ Start on GI

☐ Autoset Starttime

☒ UTC Mode

Auto TA 10

Record Select Indicators

☒ 0 ☒ 1 ☒ 2 ☒ 3

☒ 4 ☒ 5 ☒ 6 ☒ 7

☒ 8 ☒ 9 ☒ 10 ☒ 11

☒ 12 ☒ 13 ☒ 14 ☒ 15

Log File

Log File: /usr/local/log/log.xml

The final configuration is titled “Select Options”. If a scenario was selected in the previous step, the values displayed will be read from the scenario sdrr.xml file found in the /usr/local/scenarios/scenario directory. This file can be modified to reflect some of the default options above. Once all selections are made, click **Finish** to load SDRR.

These are the available options:

Time

Scenario start time. It allows you to set the start point in the scenario. The time displayed comes from the **sdrr.xml**

Date

Scenario date. Useful for multi days scenarios; it can help advancing days.

Controls

Start on GI

Starts upon receiving GI message from HOST. The HOST sends a GI message to inform attached devices, when it’s ready. SDRR will start when this message is received.

Autoset Starttime

SIRDS will determine start time based on first radar message (time stamp from the .srv files; usually used to playback live radar recordings).

UTC Mode

Synchronizes scenario to system time. It will adjust system message time to UTC format. All messages containing time substitution markers ($....$) will be adjusted.

Auto TA

SIRDS will generate TA messages in response to TI messages.

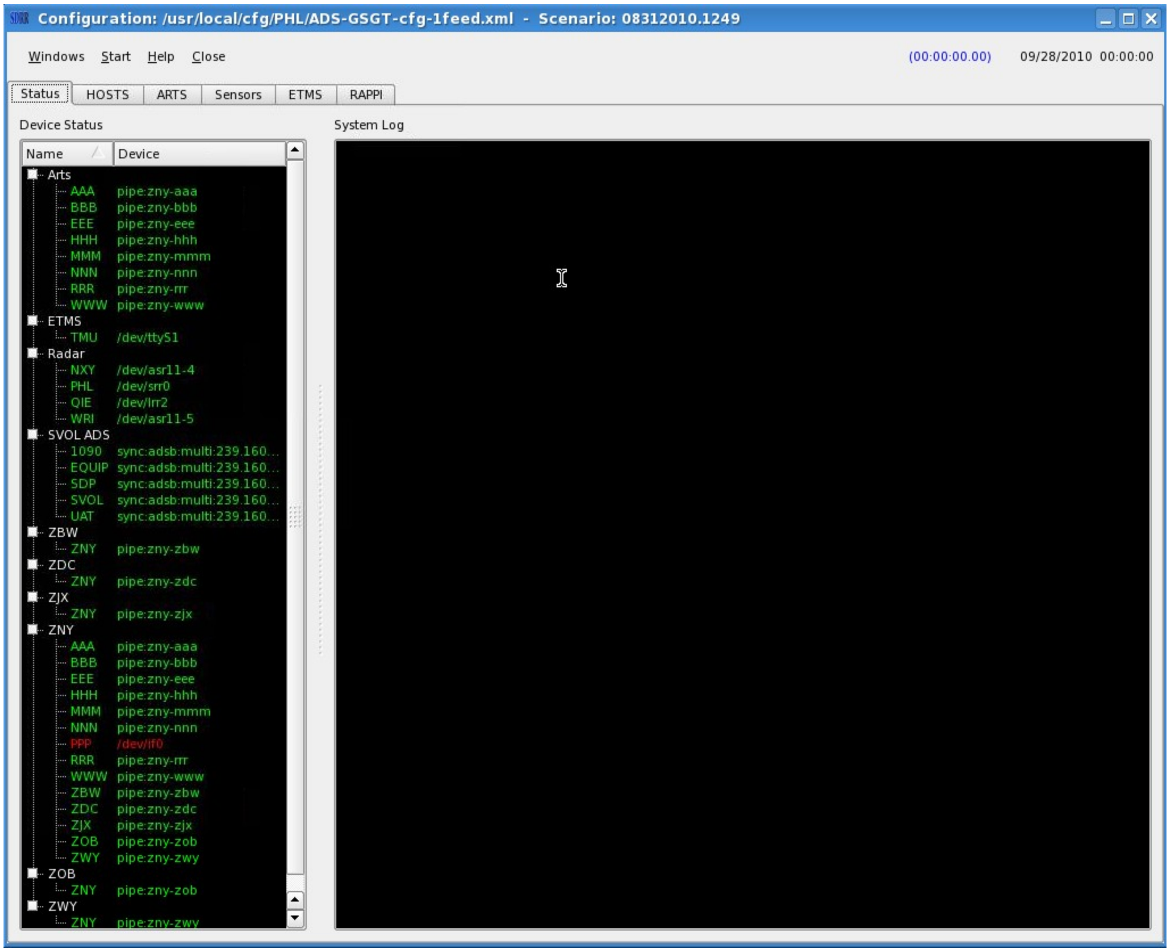
Record Select Indicators

Messages tagged with the selected indicators will be injected.

Log File

Location of the SIRDS log file (rename to the scenario related name for easier tracking).

4. Scenario Playback



Configuration: /usr/local/cfg/PHL/ADS-GSGT-cfg-1feed.xml - Scenario: 08312010.1249

Name	Device
Arts	
AAA	pipe:zny-aaa
BBB	pipe:zny-bbb
EEE	pipe:zny-eee
HHH	pipe:zny-hhh
MMM	pipe:zny-mmm
NNN	pipe:zny-nnn
RRR	pipe:zny-rrr
WWW	pipe:zny-www
ETMS	
TMU	/dev/ttyS1
Radar	
NXY	/dev/asr11-4
PHL	/dev/srr0
QIE	/dev/lrr2
WRI	/dev/asr11-5
SVOL ADS	
1090	sync:adsb:multi:239.160...
EQUIP	sync:adsb:multi:239.160...
SDP	sync:adsb:multi:239.160...
SVOL	sync:adsb:multi:239.160...
UAT	sync:adsb:multi:239.160...
ZBW	
ZNY	pipe:zny-zbw
ZDC	
ZNY	pipe:zny-zdc
ZJX	
ZNY	pipe:zny-zjx
ZNY	
AAA	pipe:zny-aaa
BBB	pipe:zny-bbb
EEE	pipe:zny-eee
HHH	pipe:zny-hhh
MMM	pipe:zny-mmm
NNN	pipe:zny-nnn
PPP	/dev/if0
RRR	pipe:zny-rrr
WWW	pipe:zny-www
ZBW	pipe:zny-zbw
ZDC	pipe:zny-zdc
ZJX	pipe:zny-zjx
ZOB	pipe:zny-zob
ZWY	pipe:zny-zwy
ZOB	
ZNY	pipe:zny-zob
ZWY	
ZNY	pipe:zny-zwy

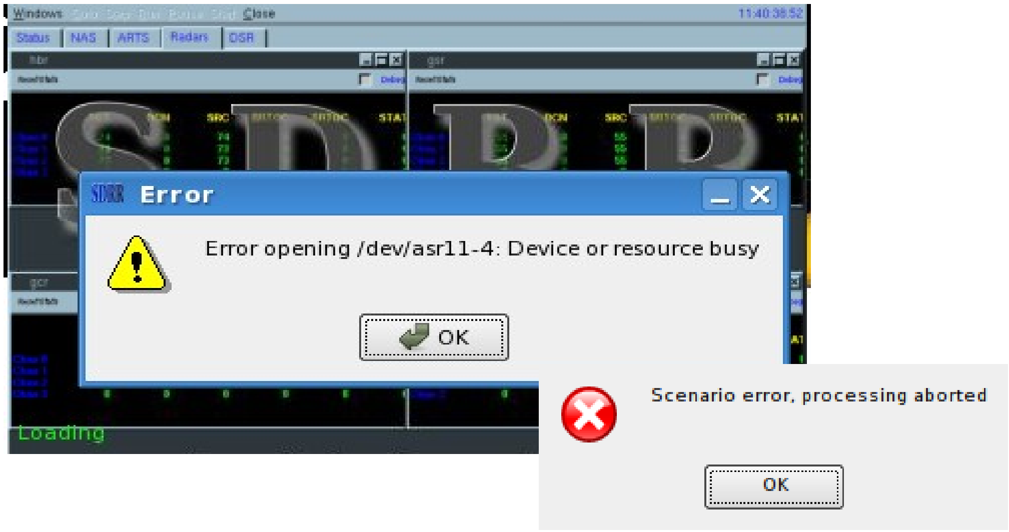
After SDRR application loads successfully, the **Status Window** appears. The left side of the status window shows **Device Status**. This is a direct reflection of the contents within the config,xml file. The white tags are expandable to show the devices/facilities within. These devices, physical or simulated, are displayed in **green**. The prefix “pipe” indicates an internal simulation. The Radars and SVOL can be expanded to list all the defined radar and the device they are assigned to.

*For example, under the Arts block AAA is shown with a pipe connection to ZNY. This indicates that ZNY and AAA are able to communicate with each other via an SDRR simulated pipe. In this case, PPP, the live site, could handoff a flight through ZNY to AAA and AAA could DA or DR back to PPP. Above, the PPP Terminal site located under the Host ZNY is the physical device **/dev/if0**. This connects ZNY to the live Terminal PPP. This link is red indicating a down link. This link should turn **Green** as soon as the scenario is opened and a clock signal is detected.*

The right side of the status window shows the **System Log.** The log displays error and warning messages about the scenario. **Red** messages indicate a problem. Not all problems will affect the success of the scenario but should be noted. A log file of each run can be found in /usr/local/log/timestamp.

The top of the window has two times and a date. The **blue** is the scenario time and the **black** is the system time. When UTC is selected or defined in sdrr.xml, the black time should show GMT time to match your ATC system.

NOTE:



Error

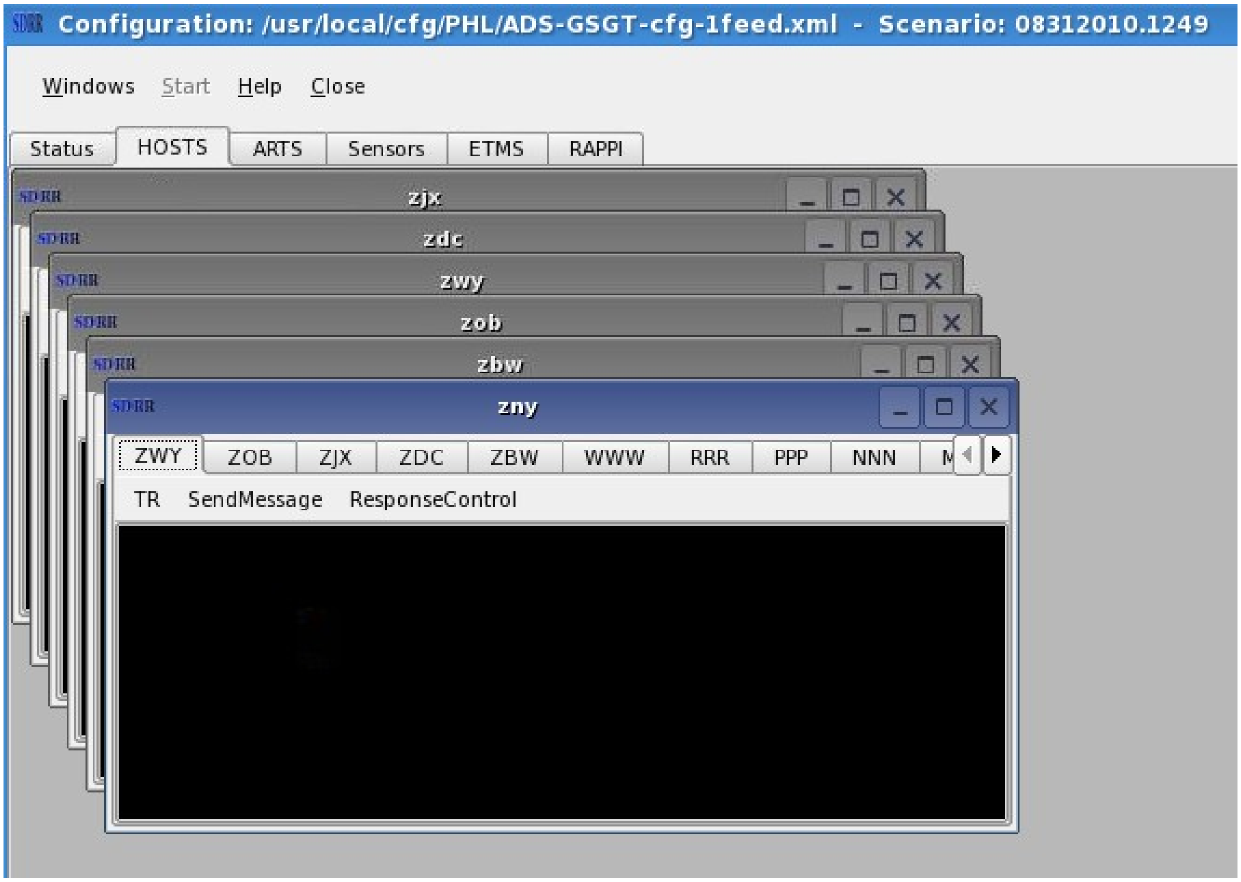
Error opening /dev/asr11-4: Device or resource busy

OK

Scenario error, processing aborted

OK

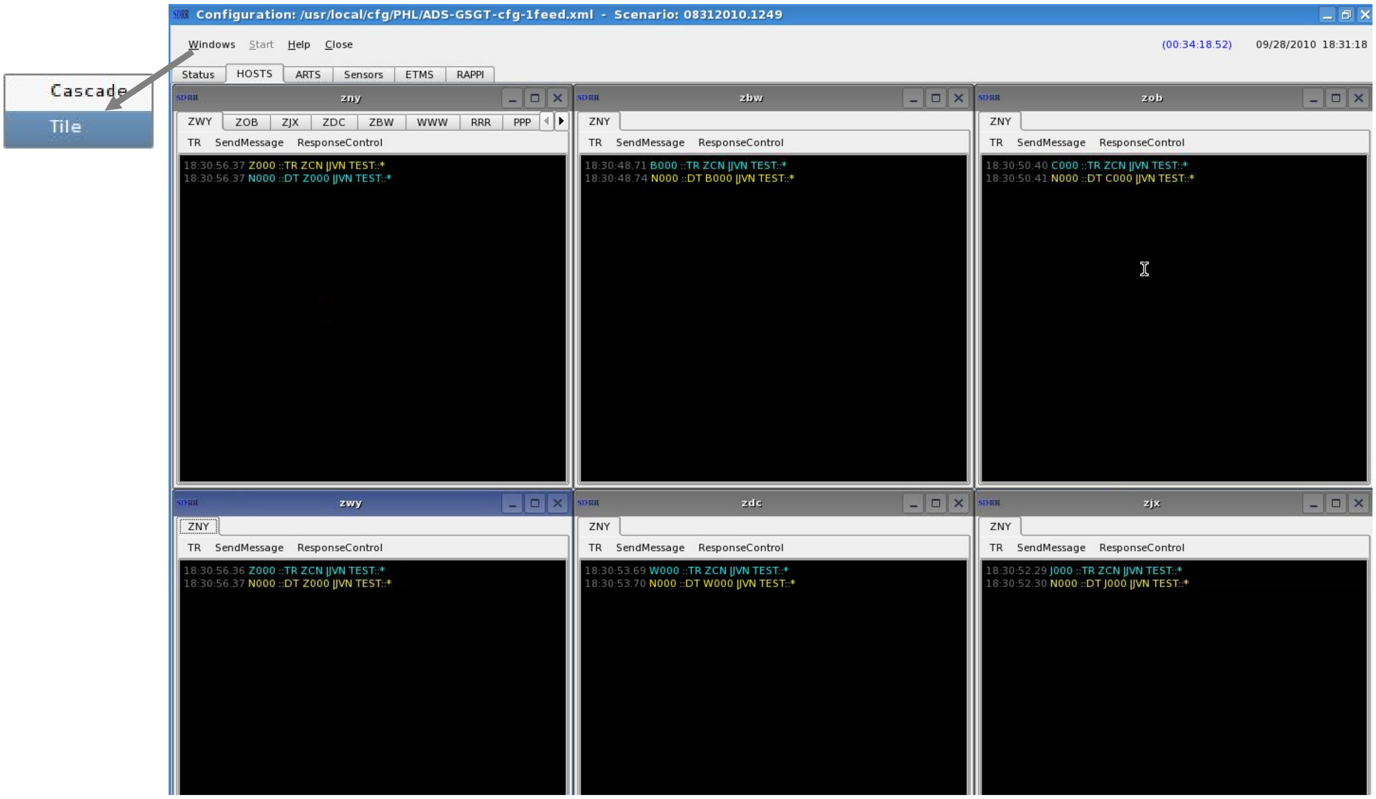
*The example above is a possible error which can occur while the scenario is loading. Read the error carefully. Chances are that the answer to the problem is being displayed. The error above is about /dev/asr11-4. This was a typological error in the ZNY-PPP.xml config file. A quick look inside the **/usr/local/cfg/config.xml** file will show the asr11-4, should have been asr11-5. After the error is corrected, the error messages should be acknowledged and SDRR restarted.*



Configuration: /usr/local/cfg/PHL/ADS-GSGT-cfg-1feed.xml - Scenario: 08312010.1249

After clicking on the **HOSTS** tab you'r able to see what centers and terminals are adapted. For a better view select Windows -> Tile (picture bellow).

*In the example below, **zny** is the central HOST. You can see ZNY is threaded to communicate with the all other facilities. This is based on the information found in the config.xml file. You can hit the TR button to send a test message to the Host. A **Blue TR** to message should go out and a **Yellow DT** response should come back. This proves the configuration file has threaded these two facilities correctly.*


Cascade
Tile
Configuration: /usr/local/cfg/PHL/ADS-GSGT-cfg-1feed.xml - Scenario: 08312010.1249
Windows Start Help Close
(00:34:18.52) 09/28/2010 18:31:18
Status HOSTS ARTS Sensors ETMS RAPPI
zny
ZWY ZOB ZJX ZDC ZBW WWW RRR PPP
TR SendMessage ResponseControl
zbw
ZNY
zob
zwy
zdc
zjx

```
18:30:56.37 Z000 ::TR ZCN |JVN TEST::*
18:30:56.37 N000 ::DT Z000 |JVN TEST::*
```

```
18:30:48.71 B000 ::TR ZCN |JVN TEST::*
18:30:48.74 N000 ::DT B000 |JVN TEST::*
```

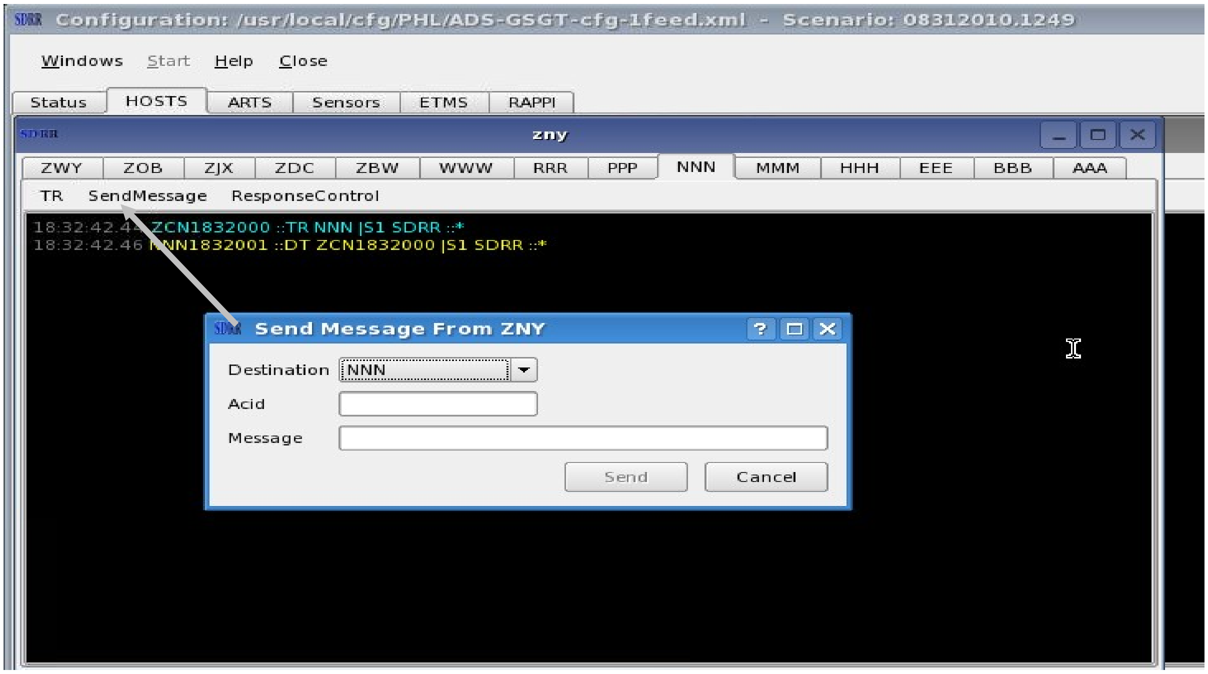
```
18:30:50.40 C000 ::TR ZCN |JVN TEST::*
18:30:50.41 N000 ::DT C000 |JVN TEST::*
```

```
18:30:56.36 Z000 ::TR ZCN |JVN TEST::*
18:30:56.37 N000 ::DT Z000 |JVN TEST::*
```

```
18:30:53.69 W000 ::TR ZCN |JVN TEST::*
18:30:53.70 N000 ::DT W000 |JVN TEST::*
```

```
18:30:52.29 J000 ::TR ZCN |JVN TEST::*
18:30:52.30 N000 ::DT J000 |JVN TEST::*
```

Under each facility you will find a **SendMessage** and **ResonseControl button**. These buttons give users the ability to send any interfacility messages as well as the ability to control how the simulated facilities respond. These messages are sent real-time on the fly. This provides huge advantages from all angles of testing.



Configuration: /usr/local/cfg/PHL/ADS-GSGT-cfg-1feed.xml - Scenario: 08312010.1249

```
18:32:42.4  ZCN1832000 ::TR NNN |S1 SDRR ::*
18:32:42.46 NNN1832001 ::DT ZCN1832000 |S1 SDRR ::*
```

Send Message From ZNY

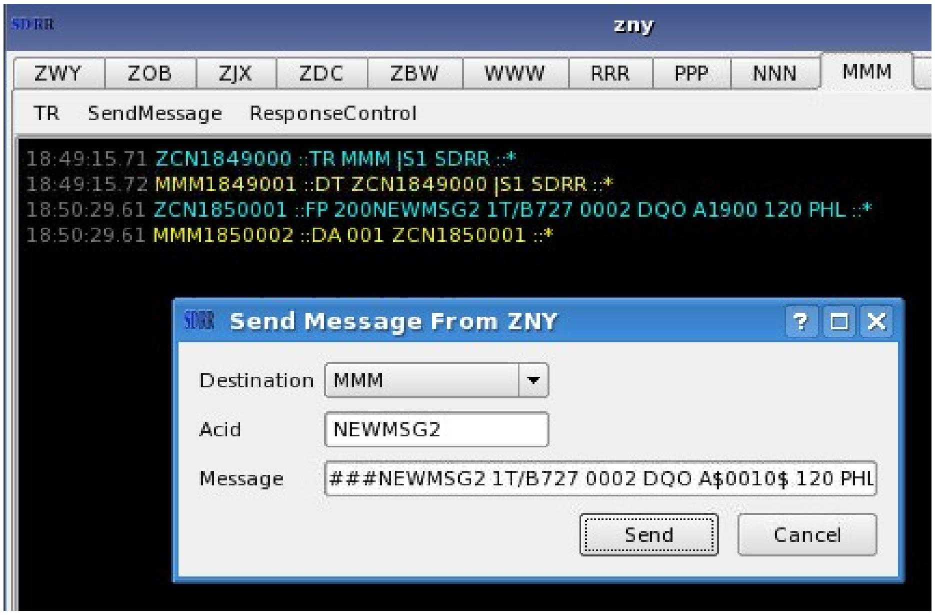
Destination NNN

Acid

Message

From the ZNY HOST tab you can select any of the listed facilities associated with ZNY. If you select the MMM window tab and select **SendMessage**, your message destination will be MMM. The example below shows how to send a flight plan from ZNY to MMM. You can also initiate messages from NON-HOST or adjacent HOSTs by selecting the desired HOST or ARTS tab, discussed later.

*From a HOSTS tab click the **SendMessage** button. The **Send Message** GUI will appear. After providing an Acid, you can begin typing your interfacility message. SDRR recognizes ### as a place to auto assign a unique ECID and @@@ as TCID. You are free to type your own ECID and TCID or let SDRR assign one. SDRR also recognizes $ at the beginning and end of the proposal time as current time. The example below will provide an arrival time of system time plus ten minutes. **NOTE; Users should only use $ substitution when the scenario start time is set to 00:00:00.** For scenarios with start times other than 00:00:00, type the exact proposal time without the $ substitution. For example, A1900 would go out just like it appears.*



```
18:49:15.71 ZCN1849000 ::TR MMM |S1 SDRR ::*
18:49:15.72 MMM1849001 ::DT ZCN1849000 |S1 SDRR ::*
18:50:29.61 ZCN1850001 ::FP 200NEWMSG2 1T/B727 0002 DQO A1900 120 PHL ::*
18:50:29.61 MMM1850002 ::DA 001 ZCN1850001 ::*
```

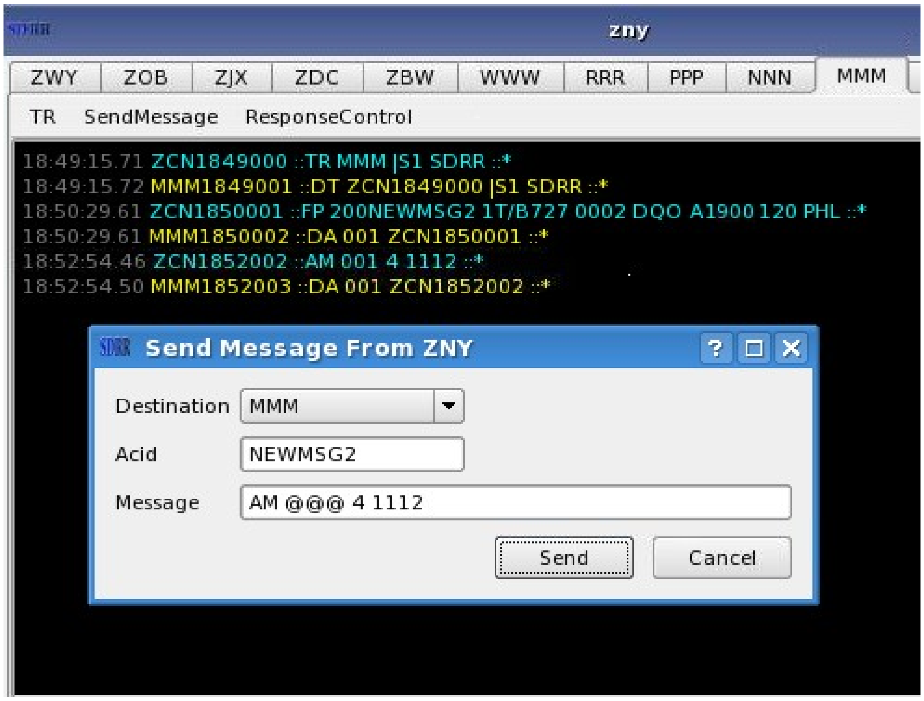
Send Message From ZNY

Destination MMM

Acid NEWMSG2

Message ###NEWMSG2 1T/B727 0002 DQO A0010 120 PHL

In the following example the beacon code is amended from 0002 to 1112. The @@@ will substitute the TCID. Notice below SDRR replaces the @@@ with 001, NEWMSG2's TCID. You can also enter the TCID.



```
18:49:15.71 ZCN1849000 ::TR MMM |S1 SDRR ::*
18:49:15.72 MMM1849001 ::DT ZCN1849000 |S1 SDRR ::*
18:50:29.61 ZCN1850001 ::FP 200NEWMSG2 1T/B727 0002 DQO A1900 120 PHL ::*
18:50:29.61 MMM1850002 ::DA 001 ZCN1850001 ::*
18:52:54.46 ZCN1852002 ::AM 001 4 1112 ::*
18:52:54.50 MMM1852003 ::DA 001 ZCN1852002 ::*
```

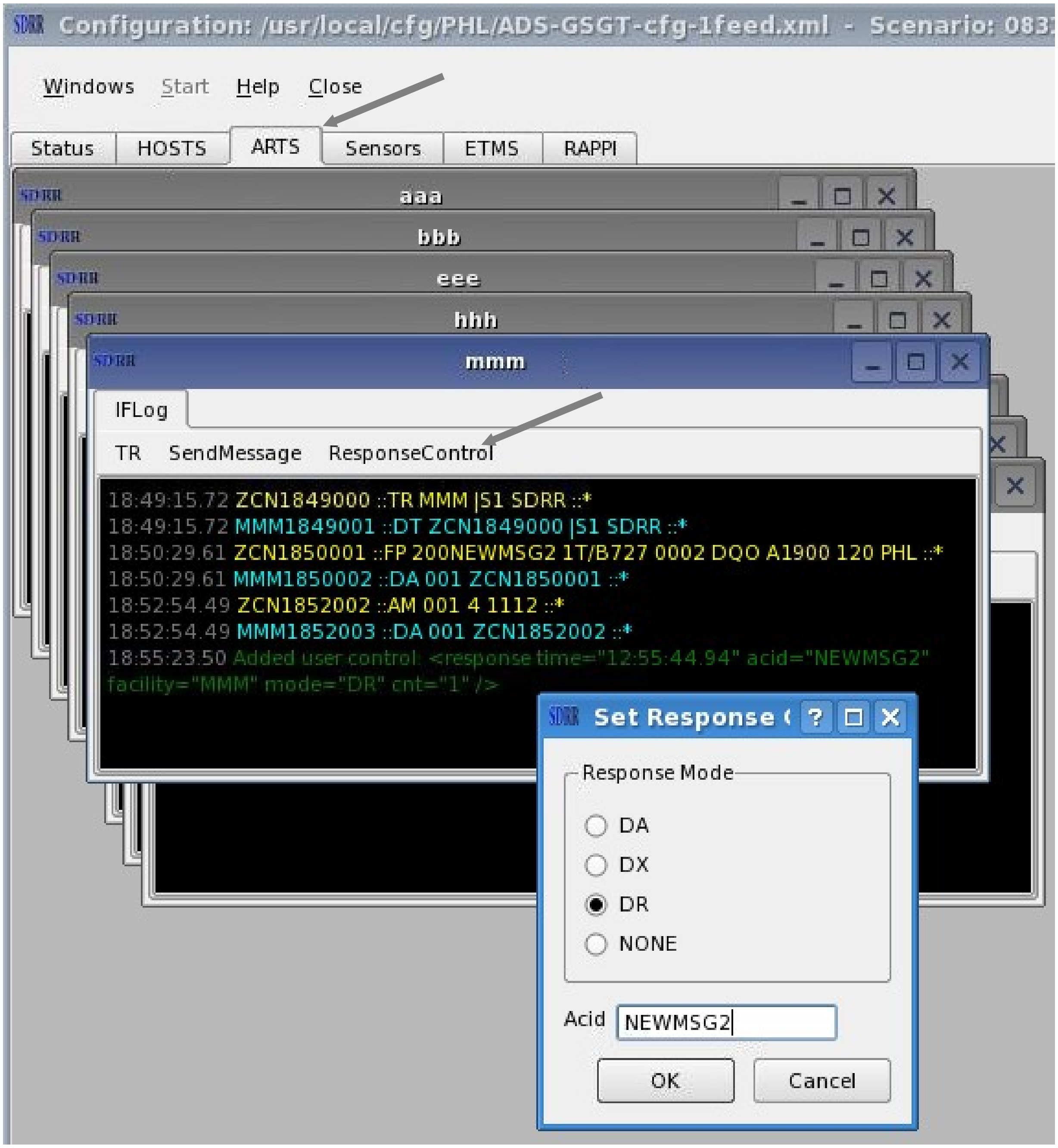
Send Message From ZNY

Destination: MMM

Acid: NEWMSG2

Message: AM @@@ 4 1112

As seen above, ZNY sent the flight plans and amendments and MMM accepted them. You can also control how the simulated facilities respond to interfacility messages. Selecting the **ResponseControl** button from a desired facility can control the selected facilities responses.



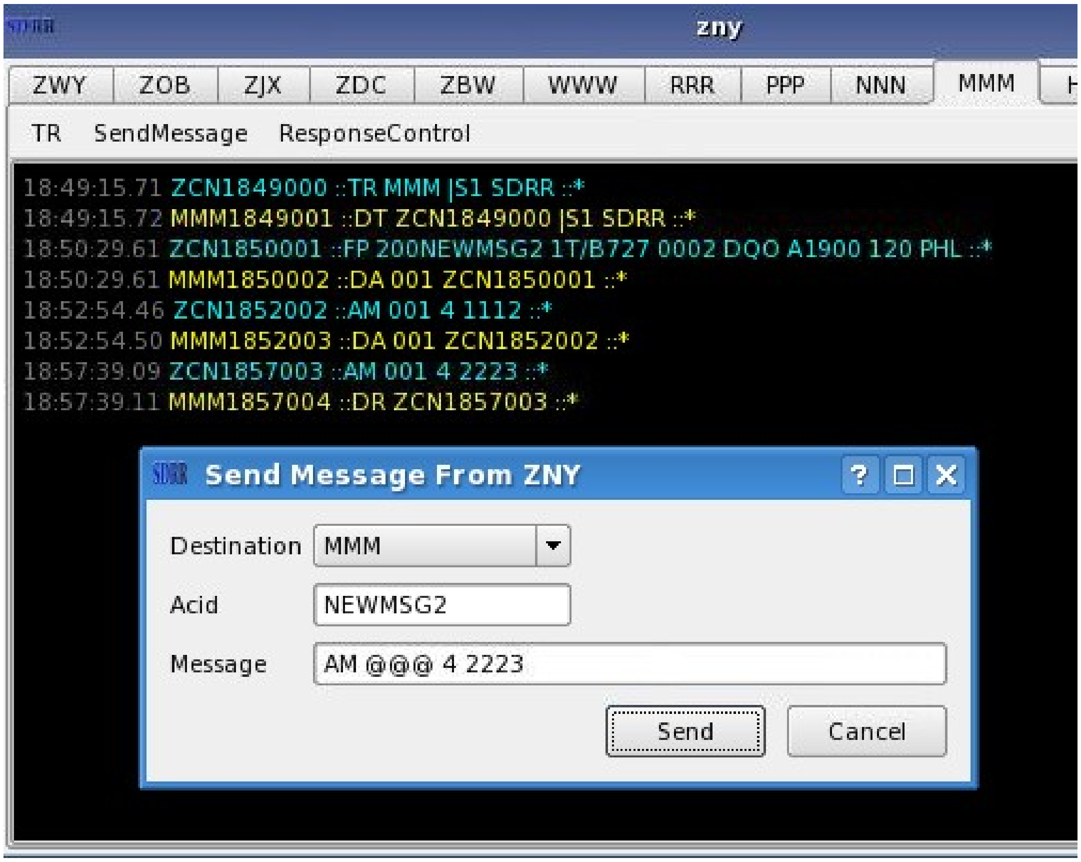
```
18:49:15.72 ZCN1849000 ::TR MMM |S1 SDRR ::*
18:49:15.72 MMM1849001 ::DT ZCN1849000 |S1 SDRR ::*
18:50:29.61 ZCN1850001 ::FP 200NEWMSG2 1T/B727 0002 DQO A1900 120 PHL ::*
18:50:29.61 MMM1850002 ::DA 001 ZCN1850001 ::*
18:52:54.49 ZCN1852002 ::AM 001 4 1112 ::*
18:52:54.49 MMM1852003 ::DA 001 ZCN1852002 ::*
18:55:23.50 Added user control: <response time="12:55:44.94" acid="NEWMSG2"
facility="MMM" mode="DR" cnt="1" />
```

Response Mode

○ DA
○ DX
● DR
○ NONE

Acid NEWMSG2

*Notice above the ARTS tab has been selected and the **ResponseControl** has been picked from the MMM facility tab window. In this example, MMM has been set to DR its next received message. Notice the green "Added user control" message above. Notice the cnt="1", this means MMM will only DR one time and return to normal processing. The green message does not go out across the interfacility line, but is displayed as a note to the user.*



```
18:49:15.71 ZCN1849000 ::TR MMM |S1 SDRR ::*
18:49:15.72 MMM1849001 ::DT ZCN1849000 |S1 SDRR ::*
18:50:29.61 ZCN1850001 ::FP 200NEWMSG2 1T/B727 0002 DQO A1900 120 PHL ::*
18:50:29.61 MMM1850002 ::DA 001 ZCN1850001 ::*
18:52:54.46 ZCN1852002 ::AM 001 4 1112 ::*
18:52:54.50 MMM1852003 ::DA 001 ZCN1852002 ::*
18:57:39.09 ZCN1857003 ::AM 001 4 2223 ::*
18:57:39.11 MMM1857004 ::DR ZCN1857003 ::*
```

Send Message From ZNY

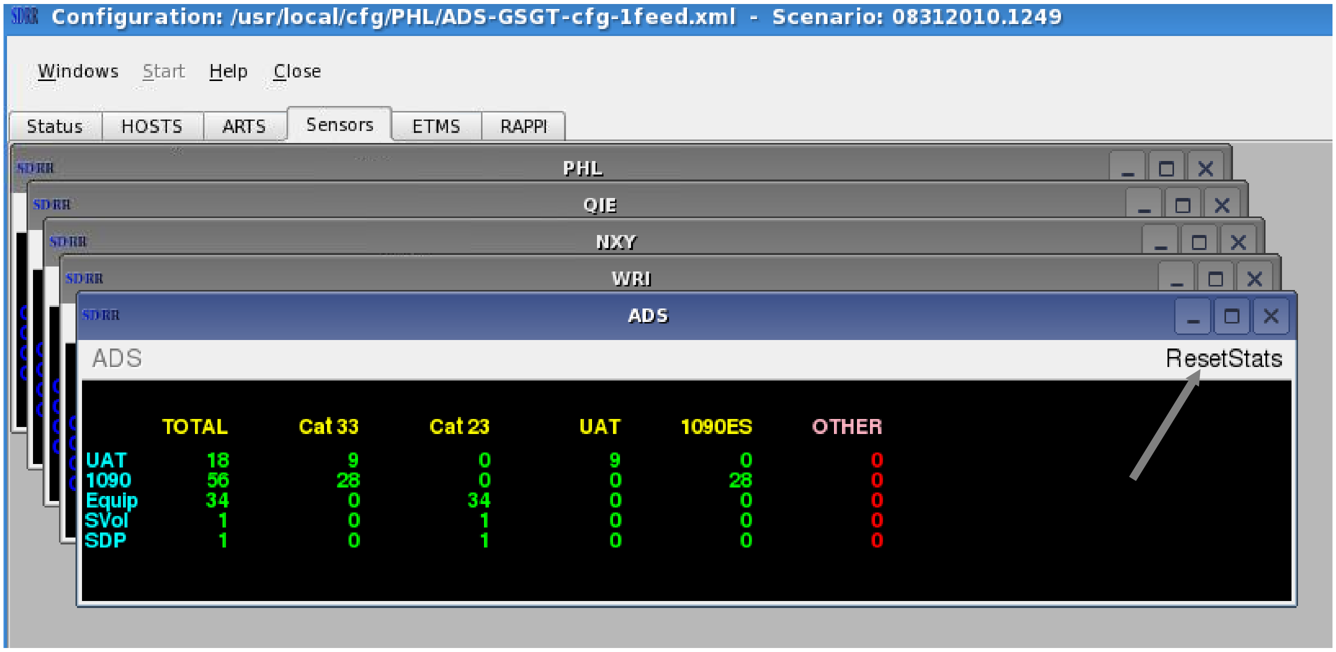
Destination: MMM

Acid: NEWMSG2

Message: AM @@@ 4 2223

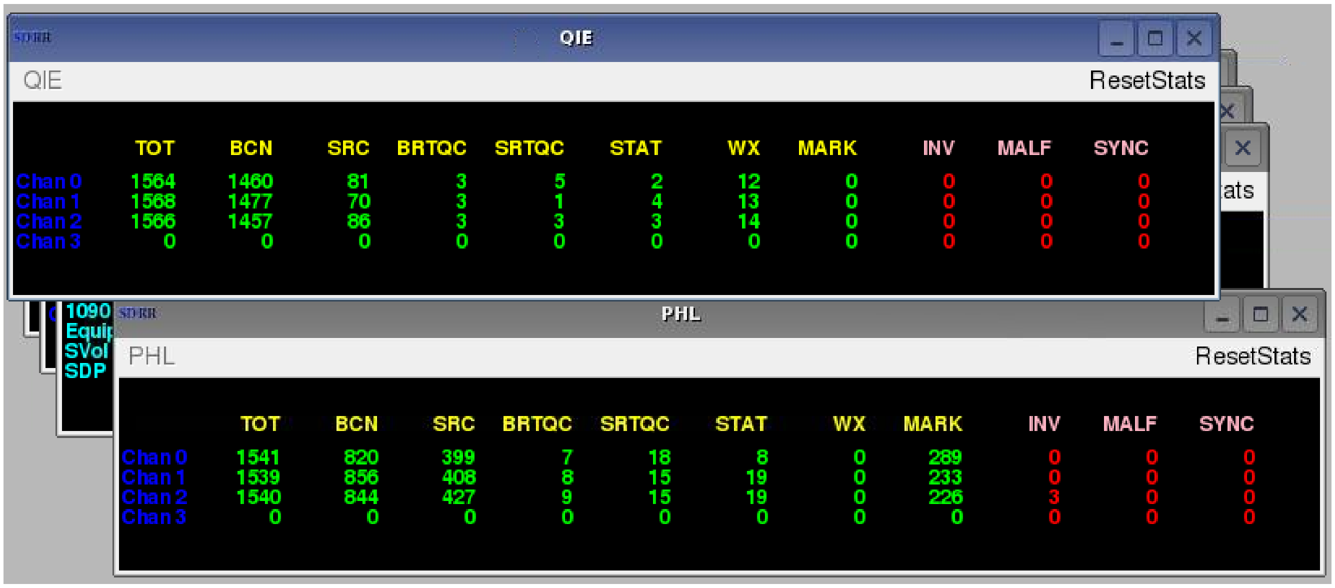
*Notice above, MMM rejected the amendment message because it had a **ResponseControl** option.*

The **Sensors** tab displays the radars defined in the configuration file. Each window shows the detail of radar channels, counts, message type and errors. The **ResetStats** button is available to reset the channel counts to zero. This does not affect the output data.


Configuration: /usr/local/cfg/PHL/ADS-GSGT-cfg-1feed.xml - Scenario: 08312010.1249

ADS

	TOTAL	Cat 33	Cat 23	UAT	1090ES	OTHER
UAT	18	9	0	9	0	0
1090	56	28	0	0	28	0
Equip	34	0	34	0	0	0
SVol	1	0	1	0	0	0
SDP	1	0	1	0	0	0



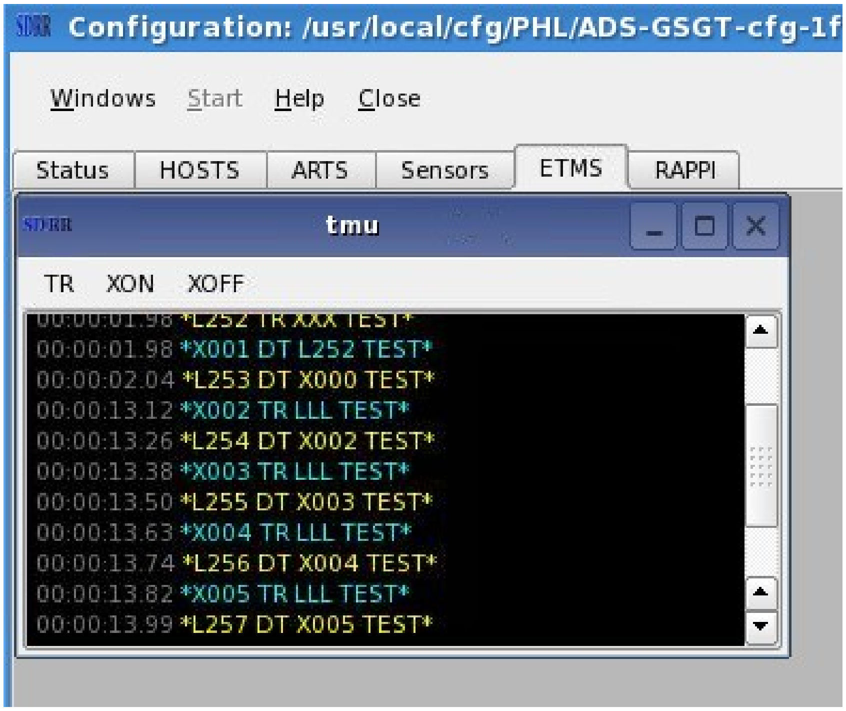
QIE

	TOT	BCN	SRC	BRTQC	SRTQC	STAT	WX	MARK	INV	MALF	SYNC
Chan 0	1564	1460	81	3	5	2	12	0	0	0	0
Chan 1	1568	1477	70	3	1	4	13	0	0	0	0
Chan 2	1566	1457	86	3	3	3	14	0	0	0	0
Chan 3	0	0	0	0	0	0	0	0	0	0	0

PHL

	TOT	BCN	SRC	BRTQC	SRTQC	STAT	WX	MARK	INV	MALF	SYNC
Chan 0	1541	820	399	7	18	8	0	289	0	0	0
Chan 1	1539	856	408	8	15	19	0	233	0	0	0
Chan 2	1540	844	427	9	15	19	0	226	3	0	0
Chan 3	0	0	0	0	0	0	0	0	0	0	0

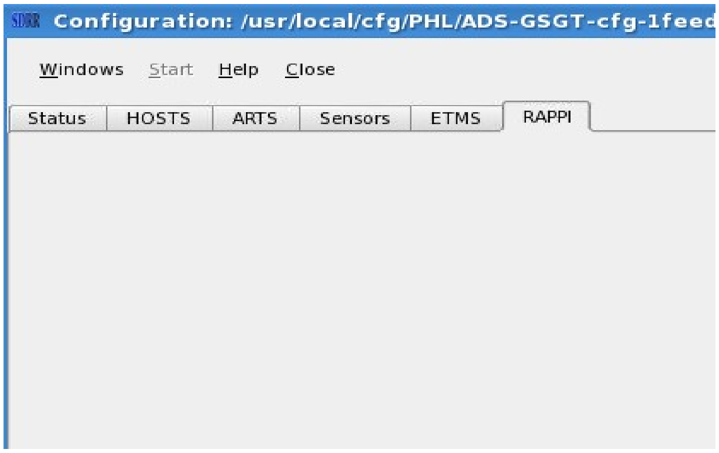
Above, you can see the QIE radar. The blue is the channel numbers and the yellow is the message type and total. The red is the error count.



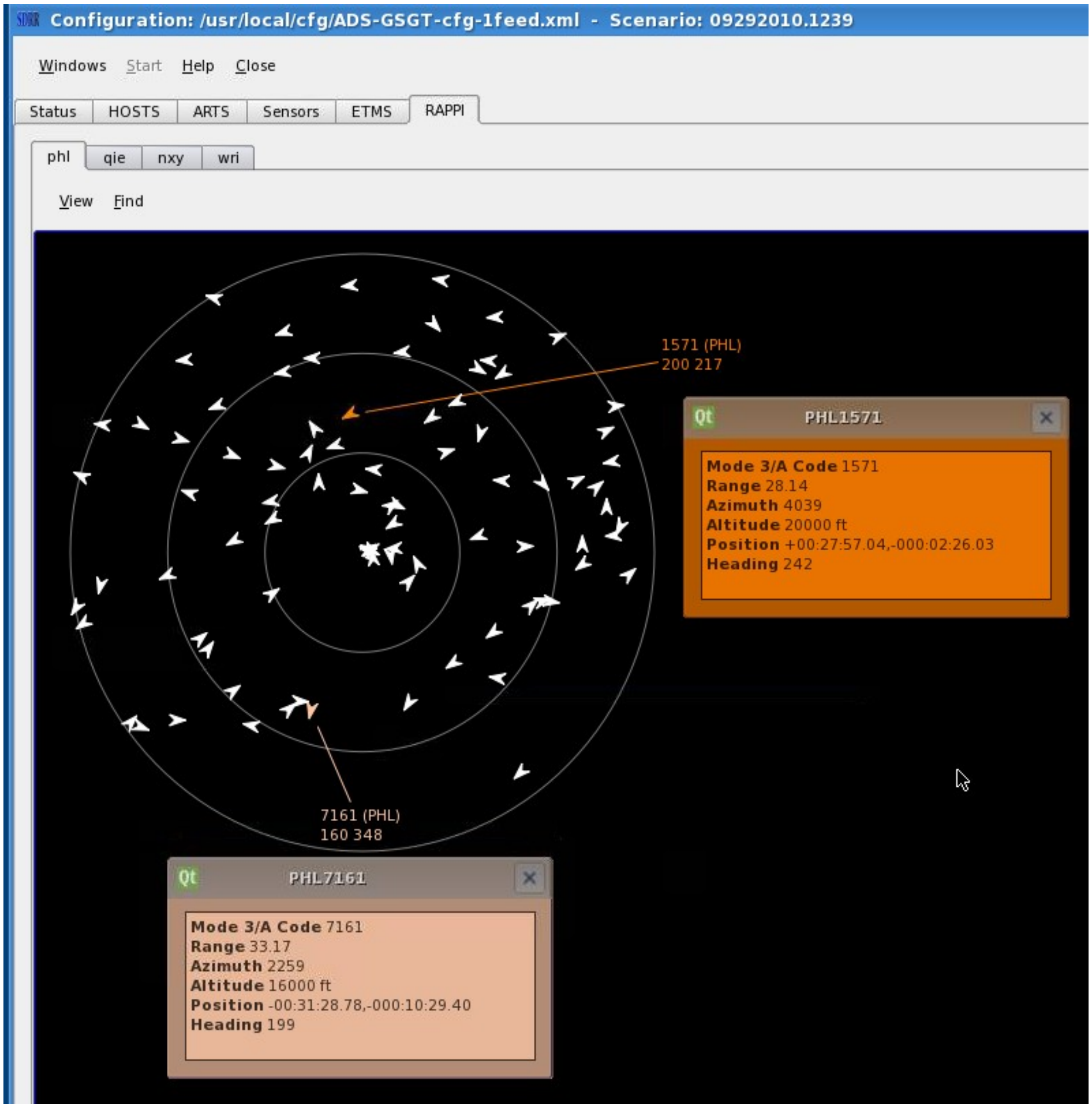
```
00:00:01.98 *L252 TR XXX TEST*
00:00:01.98 *X001 DT L252 TEST*
00:00:02.04 *L253 DT X000 TEST*
00:00:13.12 *X002 TR LLL TEST*
00:00:13.26 *L254 DT X002 TEST*
00:00:13.38 *X003 TR LLL TEST*
00:00:13.50 *L255 DT X003 TEST*
00:00:13.63 *X004 TR LLL TEST*
00:00:13.74 *L256 DT X004 TEST*
00:00:13.82 *X005 TR LLL TEST*
00:00:13.99 *L257 DT X005 TEST*
```

Above the ETMS tab has been selected. SDRR can provide an ETMS link via a DB9 serial port. Much like the interfacility data, SDRR sends blue TR / DT and should receive yellow TR / DT messages from the Terminal. The ETMS link is defined in the config.xml.

SDRR 2.4.0 and greater provides a RAPPI TAB that requires a valid AViD, (Airspace Visualization Display), license to be installed. Without the AViD software, a blank screen is displayed under the RAPPI tab



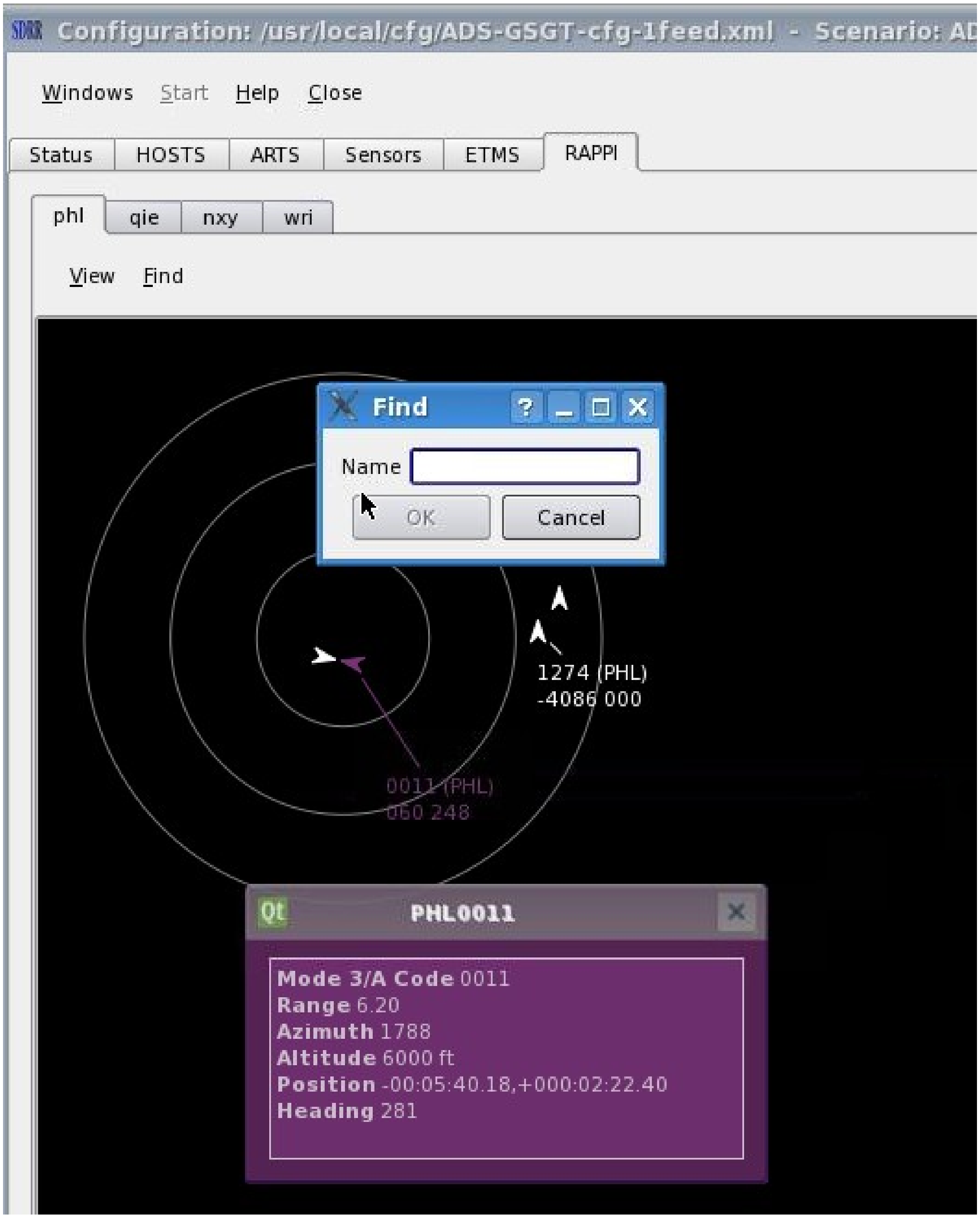
When the AViD is installed, it uses the scenarios config.xml file to configure the display, as seen below. Users can right click on targets and drag the leader line.



Mode 3/A Code 1571
Range 28.14
Azimuth 4039
Altitude 20000 ft
Position +00:27:57.04,-000:02:26.03
Heading 242

Mode 3/A Code 7161
Range 33.17
Azimuth 2259
Altitude 16000 ft
Position -00:31:28.78,-000:10:29.40
Heading 199

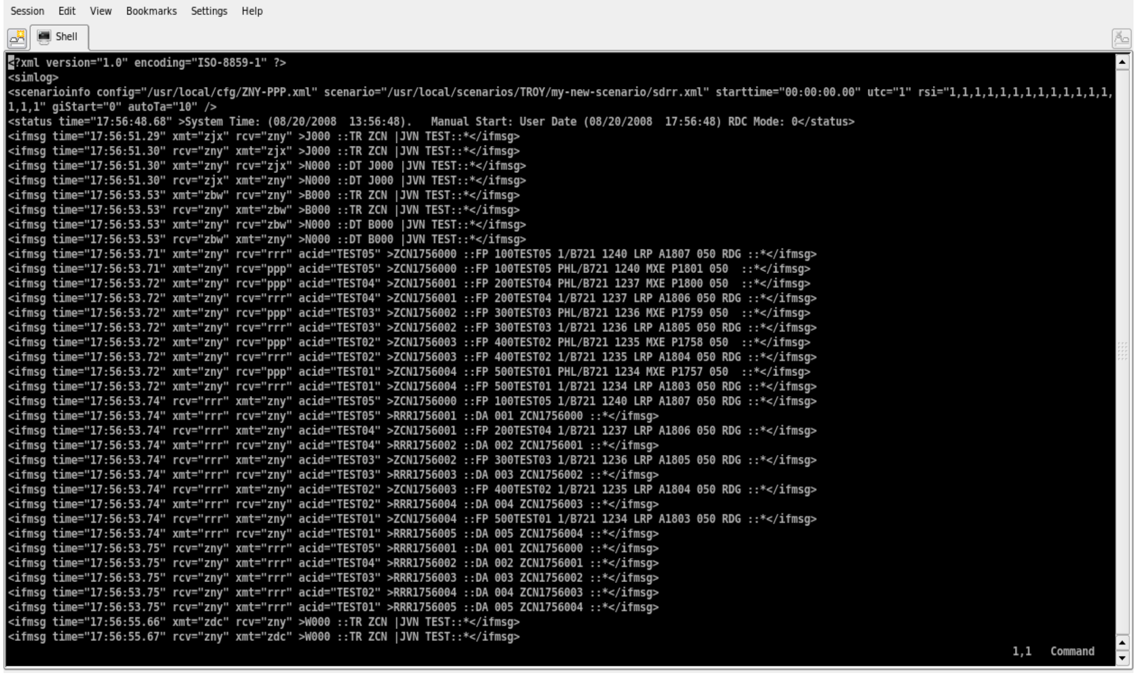
Radars found in the config.xml file will create a radar tab, as seen below. You can select the desired radar by clicking the tab. You can right click the target or click the **Find** button. You can **Find** by typing the name of the radar and a beacon code. Example, PHL0011 and click ok.



Mode 3/A Code 0011
Range 6.20
Azimuth 1788
Altitude 6000 ft
Position -00:05:40.18,+000:02:22.40
Heading 281

AViD is used for recording, displaying and analyzing data. Only a small portion of AViD features are displayed above. For more information on AViD, please refer to the AViD user manual or contact JVN.

NOTE: *This is a screenshot from the* ***log.xml*** *file. By default, the log file log.xml is located in the* ***/usr/local/log*** *directory. The messages are time-stamped so they are not overwritten. This file contains all the details from the run, including; the configuration file, the scenario, the start time, any special featured enabled and all the message transactions that took place with timestamps.*



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<simlog>
<scenarioinfo config="/usr/local/cfg/ZNY-PPP.xml" scenario="/usr/local/scenarios/TROY/my-new-scenario/sdrr.xml" starttime="00:00:00.00" utc="1" rsi="1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1" giStart="0" autoTa="10" />
<status time="17:56:48.68" >System Time: (08/20/2008  13:56:48).   Manual Start: User Date (08/20/2008  17:56:48) RDC Mode: 0</status>
<ifmsg time="17:56:51.29" xmt="zjx" rcv="zny" >J000 ::TR ZCN |JVN TEST::*</ifmsg>
<ifmsg time="17:56:51.30" rcv="zny" xmt="zjx" >J000 ::TR ZCN |JVN TEST::*</ifmsg>
<ifmsg time="17:56:51.30" xmt="zny" rcv="zjx" >N000 ::DT J000 |JVN TEST::*</ifmsg>
<ifmsg time="17:56:51.30" rcv="zjx" xmt="zny" >N000 ::DT J000 |JVN TEST::*</ifmsg>
<ifmsg time="17:56:53.53" xmt="zbw" rcv="zny" >B000 ::TR ZCN |JVN TEST::*</ifmsg>
<ifmsg time="17:56:53.53" rcv="zny" xmt="zbw" >B000 ::TR ZCN |JVN TEST::*</ifmsg>
<ifmsg time="17:56:53.53" xmt="zny" rcv="zbw" >N000 ::DT B000 |JVN TEST::*</ifmsg>
<ifmsg time="17:56:53.53" rcv="zbw" xmt="zny" >N000 ::DT B000 |JVN TEST::*</ifmsg>
<ifmsg time="17:56:53.71" xmt="zny" rcv="rrr" acid="TEST05" >ZCN1756000 ::FP 100TEST05 1/B721 1240 LRP A1807 050 RDG ::*</ifmsg>
<ifmsg time="17:56:53.71" xmt="zny" rcv="ppp" acid="TEST05" >ZCN1756000 ::FP 100TEST05 PHL/B721 1240 MXE P1801 050  ::*</ifmsg>
<ifmsg time="17:56:53.72" xmt="zny" rcv="ppp" acid="TEST04" >ZCN1756001 ::FP 200TEST04 PHL/B721 1237 MXE P1800 050  ::*</ifmsg>
<ifmsg time="17:56:53.72" xmt="zny" rcv="rrr" acid="TEST04" >ZCN1756001 ::FP 200TEST04 1/B721 1237 LRP A1806 050 RDG ::*</ifmsg>
<ifmsg time="17:56:53.72" xmt="zny" rcv="ppp" acid="TEST03" >ZCN1756002 ::FP 300TEST03 PHL/B721 1236 MXE P1759 050  ::*</ifmsg>
<ifmsg time="17:56:53.72" xmt="zny" rcv="rrr" acid="TEST03" >ZCN1756002 ::FP 300TEST03 1/B721 1236 LRP A1805 050 RDG ::*</ifmsg>
<ifmsg time="17:56:53.72" xmt="zny" rcv="ppp" acid="TEST02" >ZCN1756003 ::FP 400TEST02 PHL/B721 1235 MXE P1758 050  ::*</ifmsg>
<ifmsg time="17:56:53.72" xmt="zny" rcv="rrr" acid="TEST02" >ZCN1756003 ::FP 400TEST02 1/B721 1235 LRP A1804 050 RDG ::*</ifmsg>
<ifmsg time="17:56:53.72" xmt="zny" rcv="ppp" acid="TEST01" >ZCN1756004 ::FP 500TEST01 PHL/B721 1234 MXE P1757 050  ::*</ifmsg>
<ifmsg time="17:56:53.72" xmt="zny" rcv="rrr" acid="TEST01" >ZCN1756004 ::FP 500TEST01 1/B721 1234 LRP A1803 050 RDG ::*</ifmsg>
<ifmsg time="17:56:53.74" rcv="rrr" xmt="zny" acid="TEST05" >ZCN1756000 ::FP 100TEST05 1/B721 1240 LRP A1807 050 RDG ::*</ifmsg>
<ifmsg time="17:56:53.74" xmt="rrr" rcv="zny" acid="TEST05" >RRR1756001 ::DA 001 ZCN1756000 ::*</ifmsg>
<ifmsg time="17:56:53.74" rcv="rrr" xmt="zny" acid="TEST04" >ZCN1756001 ::FP 200TEST04 1/B721 1237 LRP A1806 050 RDG ::*</ifmsg>
<ifmsg time="17:56:53.74" xmt="rrr" rcv="zny" acid="TEST04" >RRR1756002 ::DA 002 ZCN1756001 ::*</ifmsg>
<ifmsg time="17:56:53.74" rcv="rrr" xmt="zny" acid="TEST03" >ZCN1756002 ::FP 300TEST03 1/B721 1236 LRP A1805 050 RDG ::*</ifmsg>
<ifmsg time="17:56:53.74" xmt="rrr" rcv="zny" acid="TEST03" >RRR1756003 ::DA 003 ZCN1756002 ::*</ifmsg>
<ifmsg time="17:56:53.74" rcv="rrr" xmt="zny" acid="TEST02" >ZCN1756003 ::FP 400TEST02 1/B721 1235 LRP A1804 050 RDG ::*</ifmsg>
<ifmsg time="17:56:53.74" xmt="rrr" rcv="zny" acid="TEST02" >RRR1756004 ::DA 004 ZCN1756003 ::*</ifmsg>
<ifmsg time="17:56:53.74" rcv="rrr" xmt="zny" acid="TEST01" >ZCN1756004 ::FP 500TEST01 1/B721 1234 LRP A1803 050 RDG ::*</ifmsg>
<ifmsg time="17:56:53.74" xmt="rrr" rcv="zny" acid="TEST01" >RRR1756005 ::DA 005 ZCN1756004 ::*</ifmsg>
<ifmsg time="17:56:53.75" rcv="zny" xmt="rrr" acid="TEST05" >RRR1756001 ::DA 001 ZCN1756000 ::*</ifmsg>
<ifmsg time="17:56:53.75" rcv="zny" xmt="rrr" acid="TEST04" >RRR1756002 ::DA 002 ZCN1756001 ::*</ifmsg>
<ifmsg time="17:56:53.75" rcv="zny" xmt="rrr" acid="TEST03" >RRR1756003 ::DA 003 ZCN1756002 ::*</ifmsg>
<ifmsg time="17:56:53.75" rcv="zny" xmt="rrr" acid="TEST02" >RRR1756004 ::DA 004 ZCN1756003 ::*</ifmsg>
<ifmsg time="17:56:53.75" rcv="zny" xmt="rrr" acid="TEST01" >RRR1756005 ::DA 005 ZCN1756004 ::*</ifmsg>
<ifmsg time="17:56:55.66" xmt="zdc" rcv="zny" >W000 ::TR ZCN |JVN TEST::*</ifmsg>
<ifmsg time="17:56:55.67" rcv="zny" xmt="zdc" >W000 ::TR ZCN |JVN TEST::*</ifmsg>
                                                                                                                    1,1     Command
```

5. Surveillance Recording

Please refer to the AViD user manual or contact JVN.

6. Automatic Dependant Surveillance – Broadcast (ADS-B) Support

The software releases 2.3.0 and beyond have capability to support the new ADS-B service volume definition in the configuration and sdrr.xml files. The new ‘svol’ tag supports the playback and recording of multiple streams to better model the real world. Each stream represents the data on a different UDP port. Take a look below for both cfg.xml and sdrr.xml examples: Please note, the examples below assume a route has been setup on an ether

cfg.xml :

```
<simconfig>
   <comments>ADSB test Config</comments>
   <svol name="jt0" >
    <stream name="UAT" device="multi:224.1.1.100/59900" />
    <stream name="1090" device="multi:224.1.1.100/59901" />
    <stream name="Equip" device="multi:224.1.1.100/59902" />
    <stream name="SVol" device="multi:224.1.1.100/59903" />
   </svol>
</simconfig>
```

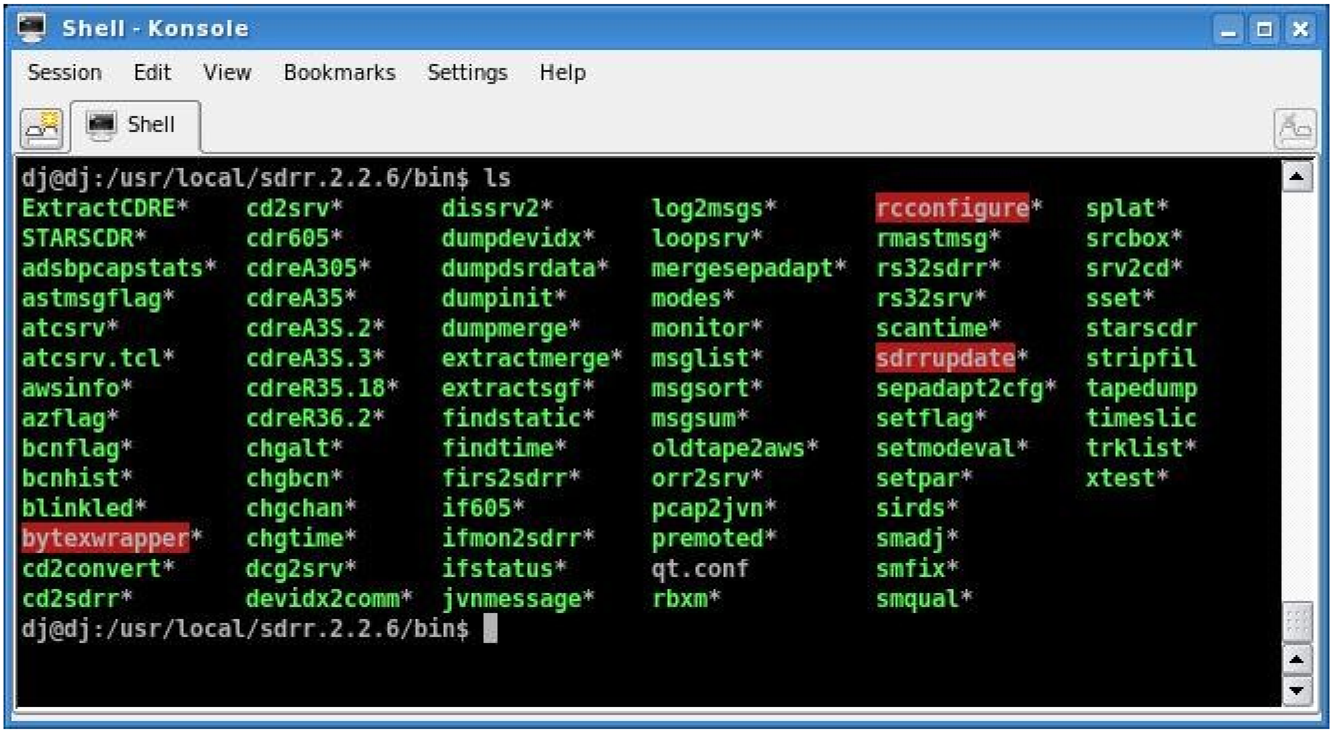
sdrr.xml :

```
<sim name="svol" starttime="20:16:46.00" giStart="0" autotu="1"
autoTa="10" utc="0" >
   <comments>Created by exportsdrr</comments>
   <svol name="at0" >
    <stream name="UAT" inast="srv/59900.ast" />
    <stream name="1090" inast="srv/59901.ast" />
    <stream name="Equip" inast="srv/59902.ast" />
    <stream name="SVol" inast="srv/59903.ast" />
   </svol>
</sim>
```

This configuration can be used for both playback and recording, which means that ADS-B data can be recorded simultaneously with radar data without using wireshark/ethereal/tcpdump programs. A recording made this way will also play back on the same individual streams. Wireshark recordings can be played back on individual streams also by running the pcap2jvn utility once for each stream specifying the UDP port and a unique file name. An sdrr.xml file can then be made by hand similar to the one above to make a scenario with the converted stream files.

7. SDRR Utilities

There are several very useful utilities which come with the SDRR installation package. They are located in the **/usr/local/sdrr.X.X.X/bin** directory. Bellow is the screenshot from the directory.



```
dj@dj:/usr/local/sdrr.2.2.6/bin$ ls
ExtractCDRE*    cd2srv*       dissrv2*       log2msgs*       rcconfigure*   splat*
STARSCDR*       cdr605*       dumpdevidx*    loopsrv*        rmastmsg*      srcbox*
adsbpcapstats*  cdreA305*     dumpdsrdata*   mergesepadapt*  rs32sdrr*      srv2cd*
astmsgflag*     cdreA35*      dumpinit*      modes*          rs32srv*       sset*
atcsrv*         cdreA35.2*    dumpmerge*     monitor*        scantime*      starscdr
atcsrv.tcl*     cdreA35.3*    extractmerge*  msglist*        sdrrupdate*    stripfil
awsinfo*        cdreR35.18*   extractsgf*    msgsort*        sepadapt2cfg*  tapedump
azflag*         cdreR36.2*    findstatic*    msgsum*         setflag*       timeslic
bcnflag*        chgalt*       findtime*      oldtape2aws*    setmodeval*    trklist*
bcnhist*        chgbcn*       firs2sdrr*     orr2srv*        setpar*        xtest*
blinkled*       chgchan*      if605*         pcap2jvn*       sirds*
bytexwrapper*   chgtime*      ifmon2sdrr*    premoted*       smadj*
cd2convert*     dcg2srv*      ifstatus*      qt.conf         smfix*
cd2sdrr*        devidx2comm*  jvnmessage*    rbxm*           smqual*
dj@dj:/usr/local/sdrr.2.2.6/bin$
```

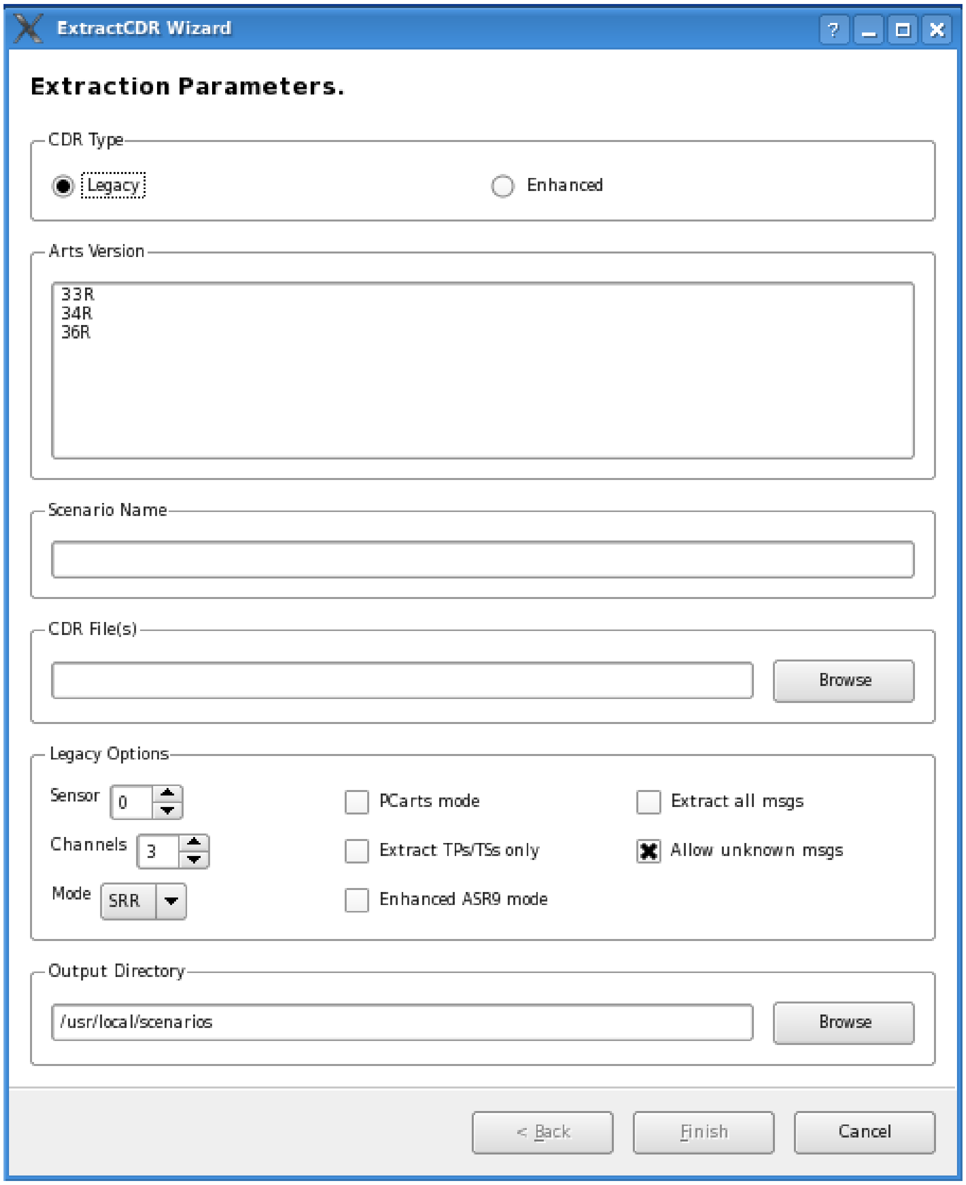
Some of the more frequently used utilities are described in the next sub-chapters.

7.1. ExtractCDR Utility

This utility extracts serial and ASTERIX data as well as interfacility messages from Common ARTS legacy & enhanced CDR files. The utility is launched by typing ***ExtractCDR*** (case sensitive) at a command prompt. This launches the ExtractCDR Wizard GUI.

7.1.1 Legacy CDR

Legacy CDR sources: Common ARTS IIe, Ie, & IIIe RGW chassis.



ExtractCDR Wizard

Extraction Parameters.

CDR Type

● Legacy ○ Enhanced

Arts Version

33R
34R
36R

Scenario Name

CDR File(s)

Legacy Options

Sensor 0

Channels 3

Mode SRR

□ PCarts mode

□ Extract TPs/TSs only

□ Enhanced ASR9 mode

□ Extract all msgs

☒ Allow unknown msgs

Output Directory

/usr/local/scenarios

Legacy CDR GUI description:

CDR type - Select either “Legacy” or “Enhanced” This section describes the extraction of “Legacy” CDR

ARTS Version - Select Common ARTS software build that produced the CDR file for extraction. (legacy CDR build versions supported: 33R, 34R, 36R)

Scenario Name – User selects directory name for SDRR scenario generated by ExtractCDR during data extraction. (i.e. “SCT080108") Select this directory in the “Scenario Selection” window during playback.

CDR File(s) - the full path where the CDR file is located (You can Browse directly to a CD/DVD drive, externally mounted USB hard drive or flash drive (legacy CDR files have a “.cdr” file extension. For example "cdr_data.cdr") **Note: Only the data file is needed for extraction do NOT select history file (cdr_data.hst)**

Options – By default, the optimal settings for most legacy CDR extractions have already been selected. The vast majority of legacy CDR comes from a IIe OSF supported site with a single 3 channel short range sensor.

Sensor - ExtractCDR will default to the first sensor (counting from zero). In the rare case of legacy data with multiple sensor input the data will have to be extracted once for each sensor and the scenario modified to combine the multiple surveillance files.
Channels - represents the number of channels created in the surveillance file. 3 channel is default. Set channel to 1 for a single channel ASR-11.
Mode - SRR (short range radar) or LRR (long range radar)

PCarts mode - no longer a valid option. To be removed in future update.

Extract TPs/TSs only - used to extract primary and secondary messages created while running in CENRAP backup mode. Used to simulate CENRAP data playback to a IIe system.

Enhanced ASR9 mode - this option will toggle the modeS bit in the CD2 serial data.

Extract all msgs - Used for the purpose of extracting TIs/TUs from interfacility CDR messages.

Allow unknown msgs - This option passes unknown message types found in various legacy Common ARTS software builds. If unchecked, ExtractCDR will fail to extract data if an unknown message type is encountered during extraction. Unknown message types are generally found in data from older Common ARTS builds and do not affect the quality of data extraction.

Output Directory - Default Output directory where the named Scenario (example "SCT080108") will be saved is /usr/local/scenarios. You may choose to save your SDRR Scenario in different location.

Click the “Finish” button after all selections have been made. A box will appear showing that data is being extracted. When complete the box will turn **green** if data has been successfully extracted. The window at the bottom of the GUI will show details of the data extraction. If the GUI turns **red**, this indicates that an error has occurred during extraction. Read the message in the details window for information regarding the failure. *See figures for examples*.

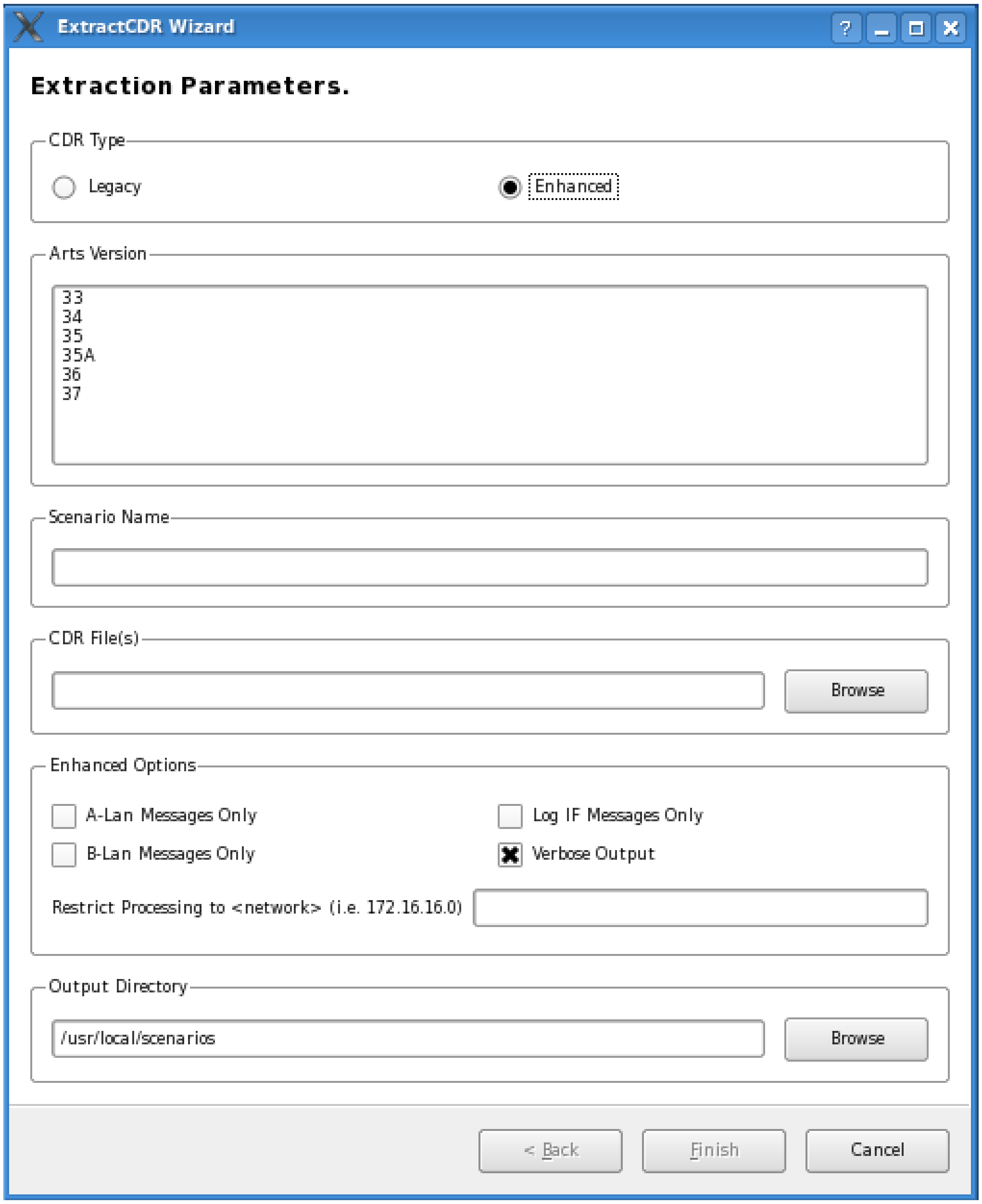
IMPORTANT NOTE: A file named **cfg.xml** will be created in the Output Directory. This file contains the device map and facility names (ARTCC & ARTS) for the site where CDR file was generated. This file needs to be copied and renamed into the **/usr/local/cfg** directory. See the following command line entry example:

```
cp /usr/local/scenarios/SCT080108/cfg.xml /usr/local/cfg/ARTS_IIeCDR_playback.xml
```

You don't have to perform this step next time when data is extracted from the same site (in our example SCT); simply select ARTS_IIeCDR_playback.xml **from the “Select Configuration” GUI during playback.**

7.1.2 Enhanced CDR

Enhanced CDR sources: Common ARTS IIIe & TAMR RAID servers



ExtractCDR Wizard

Extraction Parameters.

CDR Type

○ Legacy ● Enhanced

Arts Version

33
34
35
35A
36
37

Scenario Name

CDR File(s)

Enhanced Options

□ A-Lan Messages Only □ Log IF Messages Only
□ B-Lan Messages Only ☒ Verbose Output

Restrict Processing to <network> (i.e. 172.16.16.0)

Output Directory

/usr/local/scenarios

Enhanced CDR GUI description:

CDR type - Select either “Legacy” or “Enhanced” This section describes the extraction of “Enhanced” CDR

ARTS Version - Select Common ARTS software build that produced the CDR file for extraction. (Enhanced CDR build versions supported: 33, 34, 35, 35a, 36, 37)

Scenario Name – User selects directory name for SDRR scenario generated by ExtractCDR during data extraction. (i.e. “SCT080108")

CDR File(s) - the full path where the CDR file is located (You can Browse directly to a CD/DVD drive, externally mounted USB hard drive or flash drive (Enhanced CDR files have two forms, data copy or data merge. A data copy will have the file extension **“.dat”.** A data merge file will have the file extension **“.tso”**)

Options – By default, the optimal settings for Enhanced CDR **“merge”** extractions have already been selected.

> **A - Lan messages only -** if selected will only extract messages from Common ARTS “A” LAN
>
> **B - Lan messages only** - if selected will only extract messages from the Common ARTS “B” LAN

IMPORTANT NOTE: **If extracting data from a copy file (.dat file extension), LAN A or LAN B MUST be selected. If not duplicate messages from both LANs will be extracted. If extracting data from a “merge” file (.tso file extension), LAN A or LAN B MUST NOT be selected. Selecting a specific LAN in the case of a “merge” file will result in incomplete surveillance files.**

> **Log IF Messages Only -** A rarely used option. This option, if selected will only “log” interfacility data instead of creating proper files for interfacility message injection during playback.
>
> **Verbose Output** - this option, selected by default, produces detailed information concerning the data extracted. This information can be found in the “finished” portion of the GUI (see “details” on the following page)
>
> **Restrict processing to <network> -** unused. Designated for removal in future versions.

Output Directory - Default Output directory where the named Scenario (example "SCT080108") will be saved is /usr/local/scenarios. You may choose to save your SDRR Scenario in different location by clicking in the output directory selection box and typing your directory of choice.

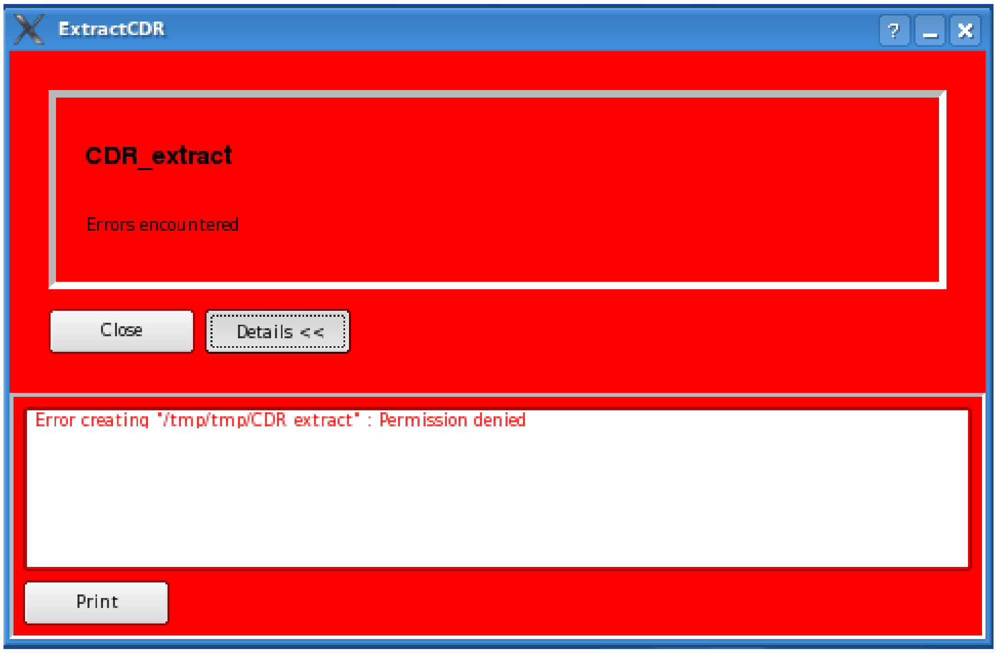
Click the “Finish” button after all selections have been made. A box will appear showing that data is being extracted. When complete the box will turn **green** if data has been successfully extracted. The window at the bottom of the GUI will show details of the data extraction. If the GUI turns **red**, this indicates that an error has occurred during extraction. Read the message in the details window for information regarding the failure. *See figures on the following page.*

IMPORTANT NOTE: A file named **cfg.xml** will be created in the Output Directory. This file contains the device map and facility names (ARTCC & ARTS) for the site where CDR file was generated. This file needs to be copied and renamed into the **/usr/local/cfg** directory. See the following command line entry example:

```
cp /usr/local/scenarios/SCT080108/cfg.xml /usr/local/cfg/SCT_CDR_playback.xml
```

You don't have to perform this step next time when data is extracted from the same site (in our example SCT); simply select SCT_CDR_playback.xml from the “Select Configuration” GUI during playback.

CDR Data extraction results

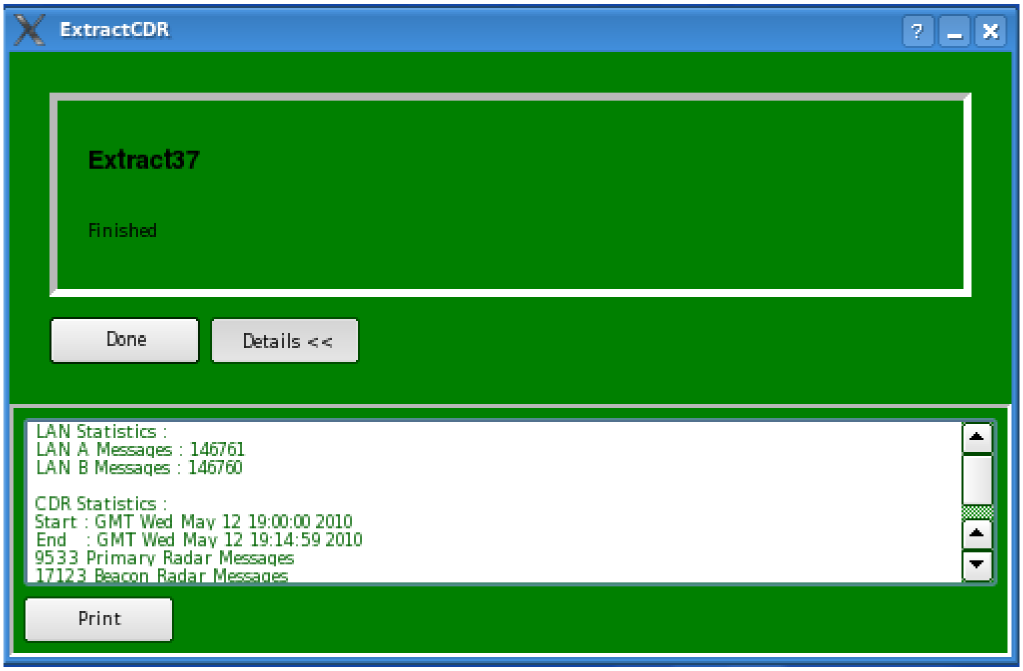


CDR_extract

Errors encountered

```
Error creating "/tmp/tmp/CDR extract" : Permission denied
```

Failed Extraction due to lack of write permissions



Extract37

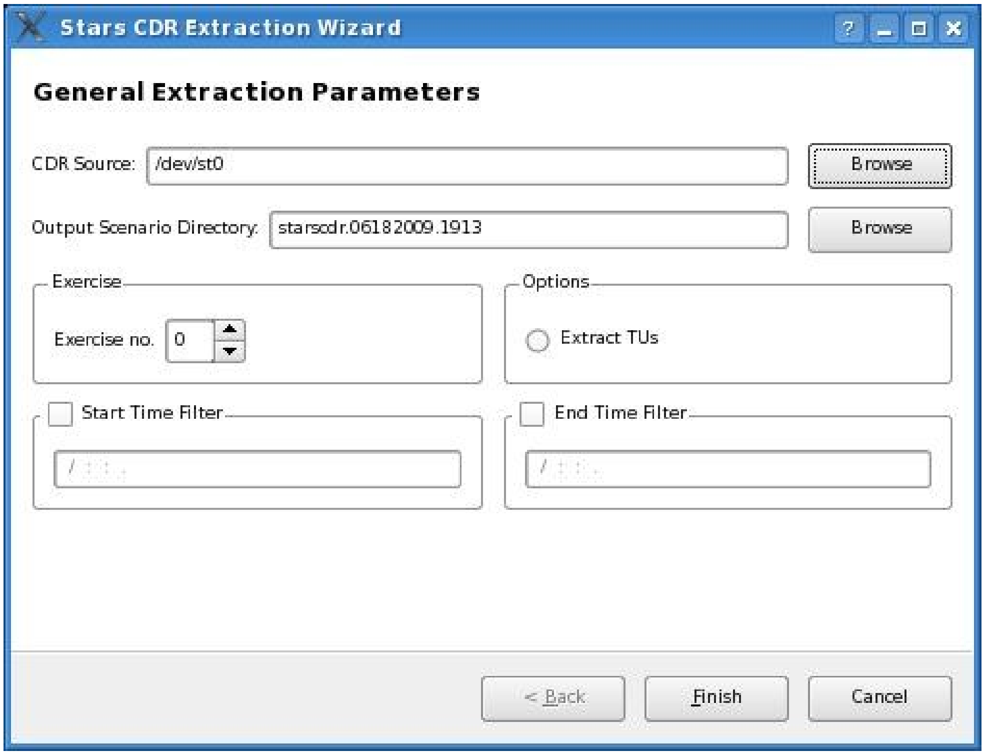
Finished

```
LAN Statistics :
LAN A Messages : 146761
LAN B Messages : 146760

CDR Statistics :
Start : GMT Wed May 12 19:00:00 2010
End   : GMT Wed May 12 19:14:59 2010
9533 Primary Radar Messages
17123 Beacon Radar Messages
```

Successful Extraction

7.2. STARSCDR Utility

Similar to the one described in 6.1., STARSCDR is used to extracts and playbacks data from STARS-CDR file. The utility is launched by typing ***STARSCDR*** (case sensitive) at a command line:



Stars CDR Extraction Wizard

General Extraction Parameters

CDR Source: /dev/st0

Output Scenario Directory: starscdr.06182009.1913

Exercise

Exercise no. 0

Options

○ Extract TUs

□ Start Time Filter

/ : : .

□ End Time Filter

/ : : .

CDR Source is the location of the CDR file, which can also be on a CD or DVD. Defaults to ***/dev/st0***, (tape drive device). You can also browse for a .cdr file with the Browse button or simply type the path in the space provided.

Output Scenario Directory is the directory where the extracted scenario will be saved. By default, the scenario will be placed in ***/usr/local/scenario/*** with the name ***starscdr.date+time***.

Exercise no. defaults to zero and is most likely correct. Exercise is referring to the mode the STARS was in while recording CDR. When CDR recording is done for special "TEST MODE" or ATCOACH runs, Exercise 1 or 2 may be needed. Exercise 0 is for normal operation/live mode.

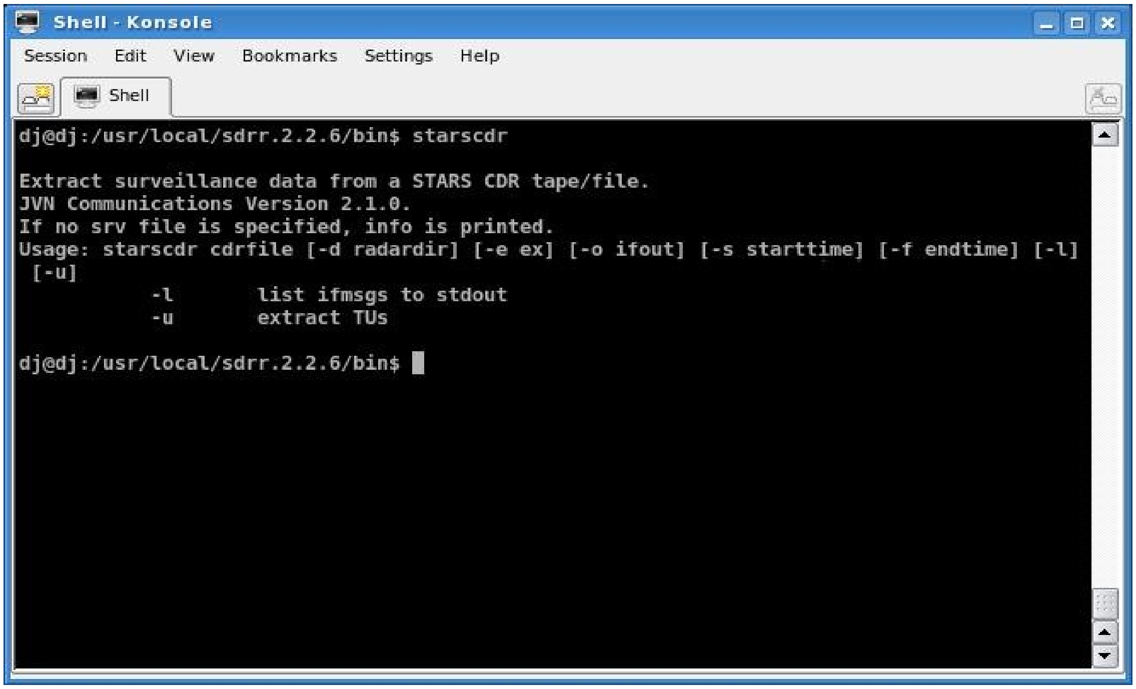
Options Extract TUs: By default STARSCDR extracts FP and AM messages only. During playback, SDRR handles the TI/TU messages based on positions and populates the position/velocity fields. For exact scenario playback, select the Extract TUs button. This will provide SDRR will the handoff positions track updated (TU) positions. Most scenario playback is done without selecting the TUs option.

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 firs 30 minutes; you can set the end time to 30 and create a small SDRR scenario.

Click the finish button when ready to extract.

Next you will see a progress window while you data is being extracted, followed by a GREEN, (Success) or RED (Failed) window. You can select the Details button for extractions details, such as Rev# site ID and message count. Details can also help to determine why an extraction may have failed

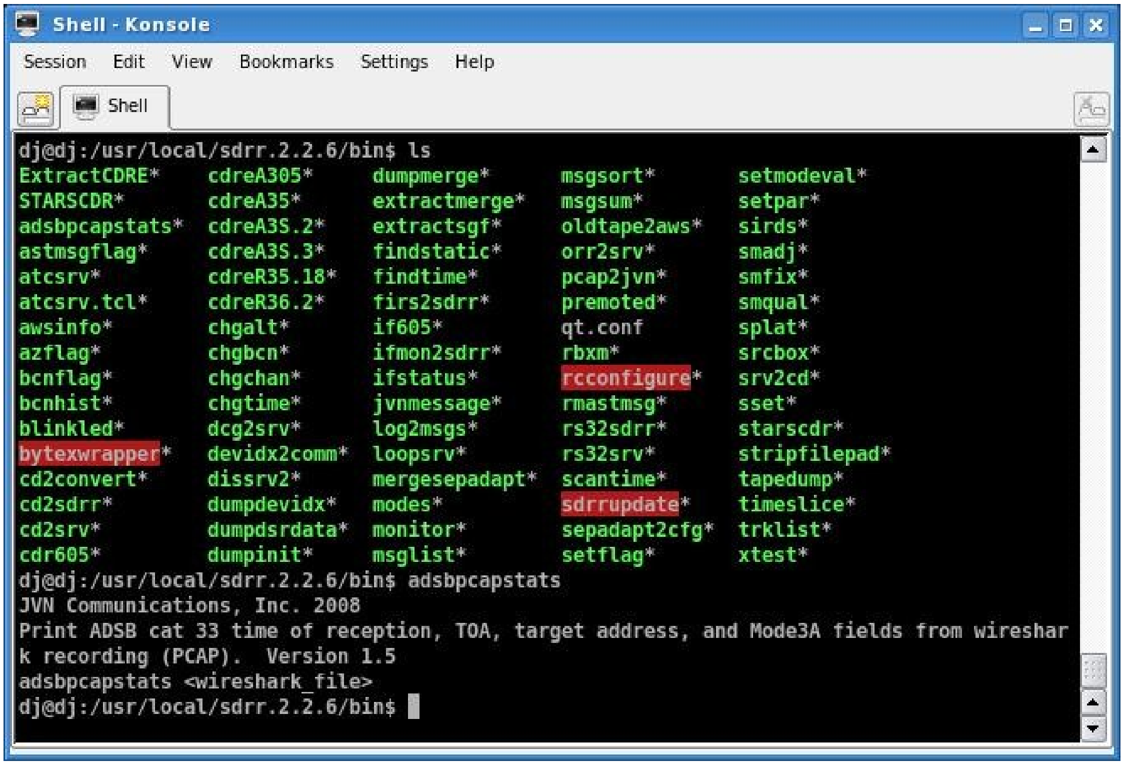
As shown below, this utility can also be executed directly from command line by typing ***starscdr cdrfile option(s)***:



```
dj@dj:/usr/local/sdrr.2.2.6/bin$ starscdr

Extract surveillance data from a STARS CDR tape/file.
JVN Communications Version 2.1.0.
If no srv file is specified, info is printed.
Usage: starscdr cdrfile [-d radardir] [-e ex] [-o ifout] [-s starttime] [-f endtime] [-l]
 [-u]
           -l       list ifmsgs to stdout
           -u       extract TUs

dj@dj:/usr/local/sdrr.2.2.6/bin$
```

7.3. adsbpcapstats Utility

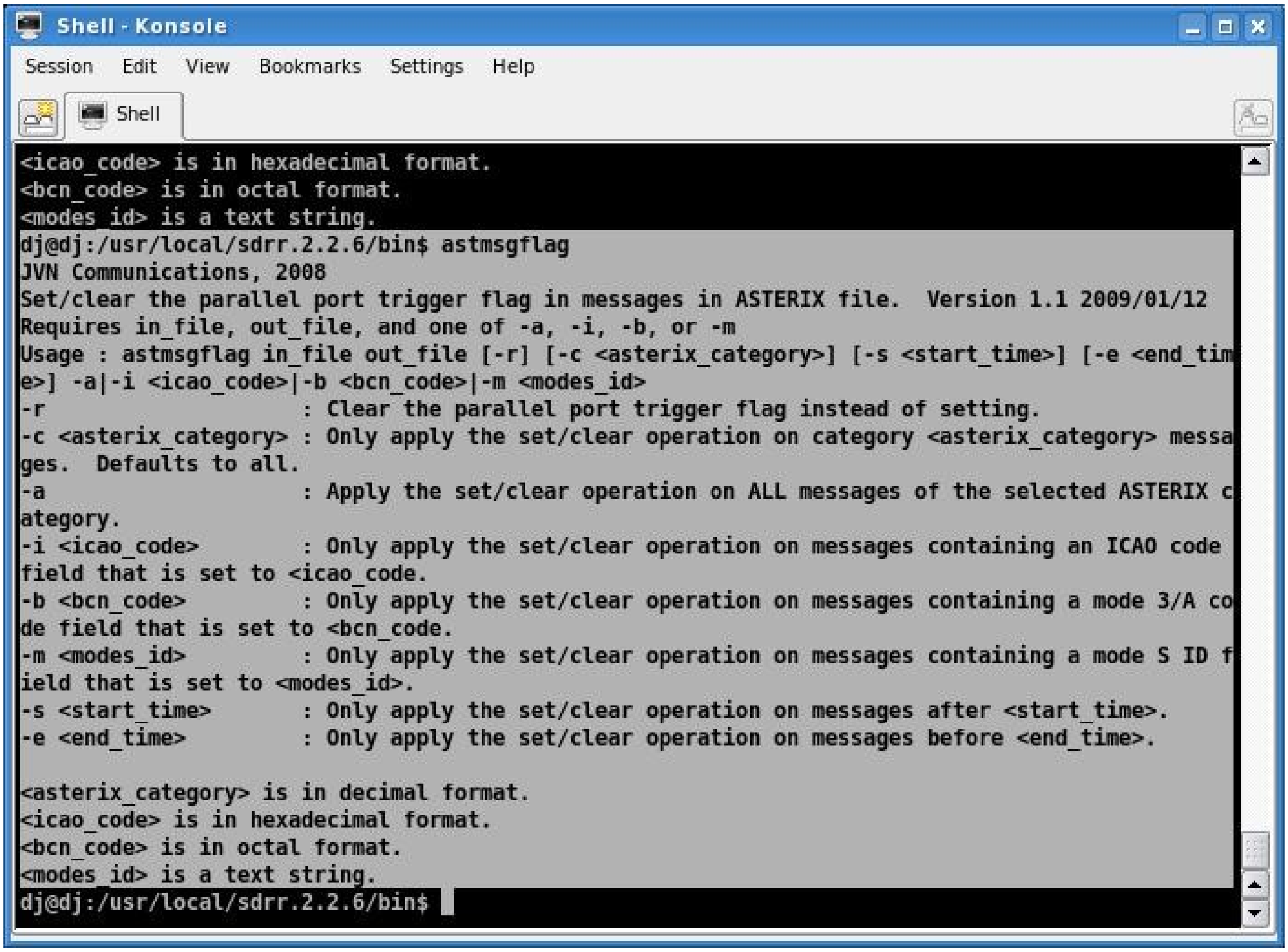
This is a utility used with modified version of the open source network analyzing application named *Wireshark*. The utility displays various statistical information for the ADSB targets captured by *Wireshark* application.



```
dj@dj:/usr/local/sdrr.2.2.6/bin$ ls
ExtractCDRE*    cdreA305*     dumpmerge*      msgsort*       setmodeval*
STARSCDR*       cdreA35*      extractmerge*   msgsum*        setpar*
adsbpcapstats*  cdreA3S.2*    extractsgf*     oldtape2aws*   sirds*
astmsgflag*     cdreA3S.3*    findstatic*     orr2srv*       smadj*
atcsrv*         cdreR35.18*   findtime*       pcap2jvn*      smfix*
atcsrv.tcl*     cdreR36.2*    firs2sdrr*      premoted*      smqual*
awsinfo*        chgalt*       if605*          qt.conf        splat*
azflag*         chgbcn*       ifmon2sdrr*     rbxm*          srcbox*
bcnflag*        chgchan*      ifstatus*       rcconfigure*   srv2cd*
bcnhist*        chgtime*      jvnmessage*     rmastmsg*      sset*
blinkled*       dcg2srv*      log2msgs*       rs32sdrr*      starscdr*
bytexwrapper*   devidx2comm*  loopsrv*        rs32srv*       stripfilepad*
cd2convert*     dissrv2*      mergesepadapt*  scantime*      tapedump*
cd2sdrr*        dumpdevidx*   modes*          sdrrupdate*    timeslice*
cd2srv*         dumpdsrdata*  monitor*        sepadapt2cfg*  trklist*
cdr605*         dumpinit*     msglist*        setflag*       xtest*
dj@dj:/usr/local/sdrr.2.2.6/bin$ adsbpcapstats
JVN Communications, Inc. 2008
Print ADSB cat 33 time of reception, TOA, target address, and Mode3A fields from wireshar
k recording (PCAP).  Version 1.5
adsbpcapstats <wireshark_file>
dj@dj:/usr/local/sdrr.2.2.6/bin$
```

7.4. astmsgflag, azflag, bcnflag Utilities

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



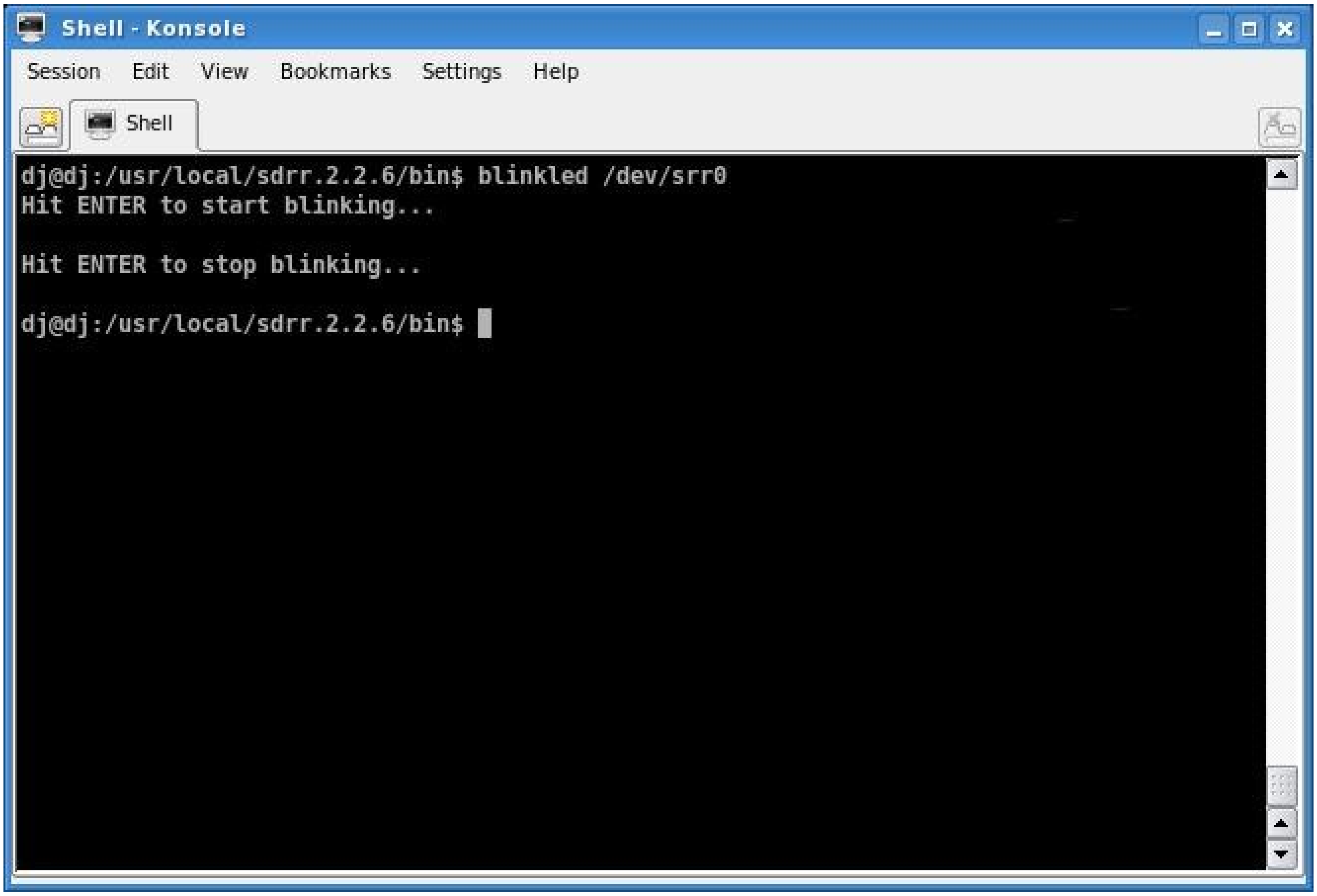
```
<icao_code> is in hexadecimal format.
<bcn_code> is in octal format.
<modes_id> is a text string.
dj@dj:/usr/local/sdrr.2.2.6/bin$ astmsgflag
JVN Communications, 2008
Set/clear the parallel port trigger flag in messages in ASTERIX file.  Version 1.1 2009/01/12
Requires in_file, out_file, and one of -a, -i, -b, or -m
Usage : astmsgflag in_file out_file [-r] [-c <asterix_category>] [-s <start_time>] [-e <end_tim
e>] -a|-i <icao_code>|-b <bcn_code>|-m <modes_id>
-r                     : Clear the parallel port trigger flag instead of setting.
-c <asterix_category> : Only apply the set/clear operation on category <asterix_category> messa
ges.  Defaults to all.
-a                     : Apply the set/clear operation on ALL messages of the selected ASTERIX c
ategory.
-i <icao_code>         : Only apply the set/clear operation on messages containing an ICAO code
field that is set to <icao_code.
-b <bcn_code>          : Only apply the set/clear operation on messages containing a mode 3/A co
de field that is set to <bcn_code.
-m <modes_id>          : Only apply the set/clear operation on messages containing a mode S ID f
ield that is set to <modes_id>.
-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>.

<asterix_category> is in decimal format.
<icao_code> is in hexadecimal format.
<bcn_code> is in octal format.
<modes_id> is a text string.
dj@dj:/usr/local/sdrr.2.2.6/bin$
```

7.5. atcsrv and atcsrv.tcl Utilities

ATCSRV can be used via SDRR to sniff multicast feeds from ATCOACH and convert to serial data outputs. SDRR configuration files are needed to prepare network devices and serial devices.

7.6. blinkled Utility

This utility is used to identify the physical location of the surveillance and interfacility devices on the SRDD by blinking the led light on the PCI Emulex Card. The usage is as follows:



```
dj@dj:/usr/local/sdrr.2.2.6/bin$ blinkled /dev/srr0
Hit ENTER to start blinking...

Hit ENTER to stop blinking...

dj@dj:/usr/local/sdrr.2.2.6/bin$ 
```

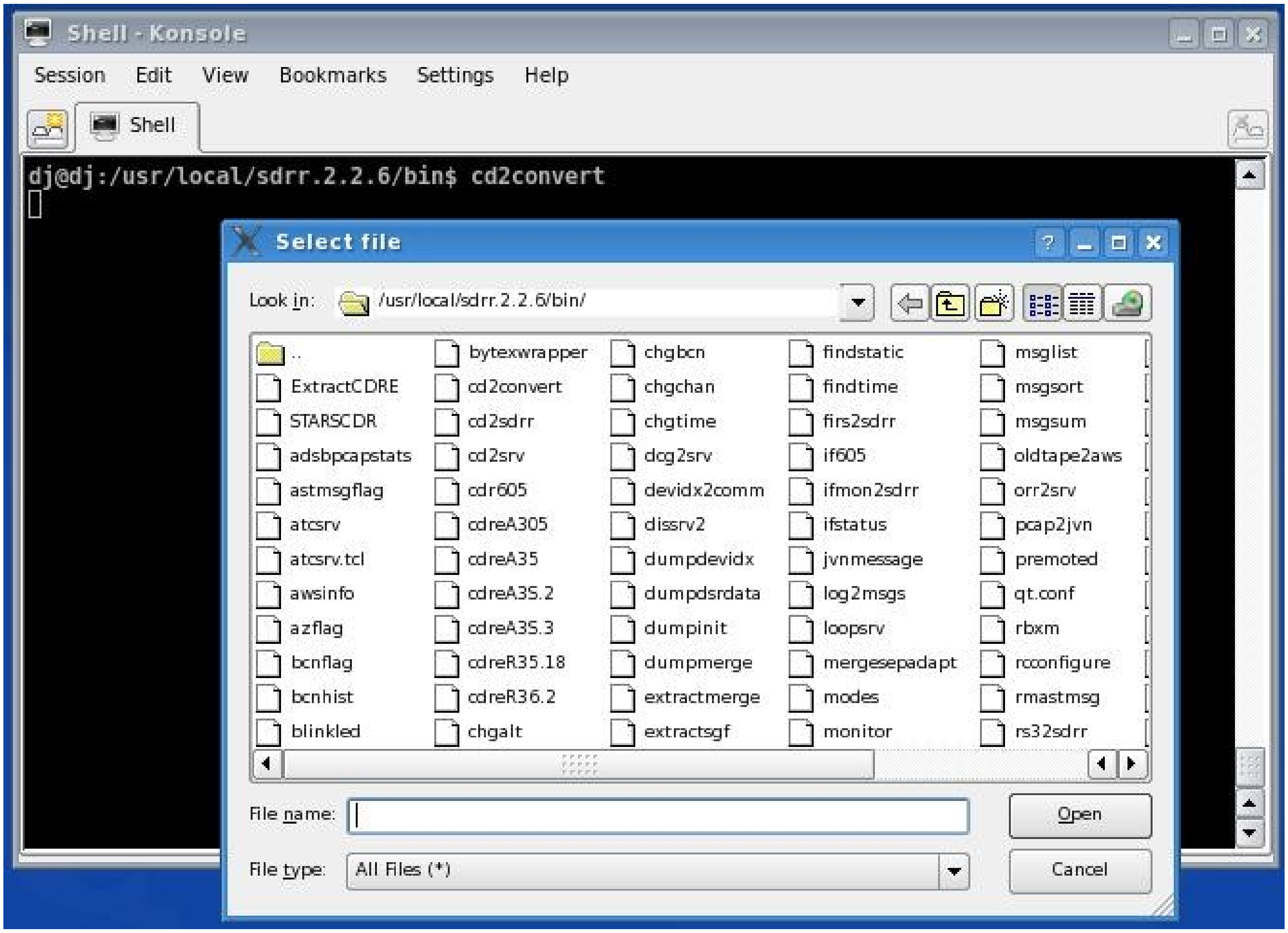
blinkled device (short range radar 0 (***srr0***) in our example)

Hit ENTER to start blinking – the led light on the ***device*** should start to blink.

Hit ENTER to stop blinking.

7.7. cd2convert, cd2sdrr and cd2srv Utilities

CD2 utilities can used to convert CD2 format data to SDRR (srv) format data. Below is a screenshot from one of them.



```
dj@dj:/usr/local/sdrr.2.2.6/bin$ cd2convert
```

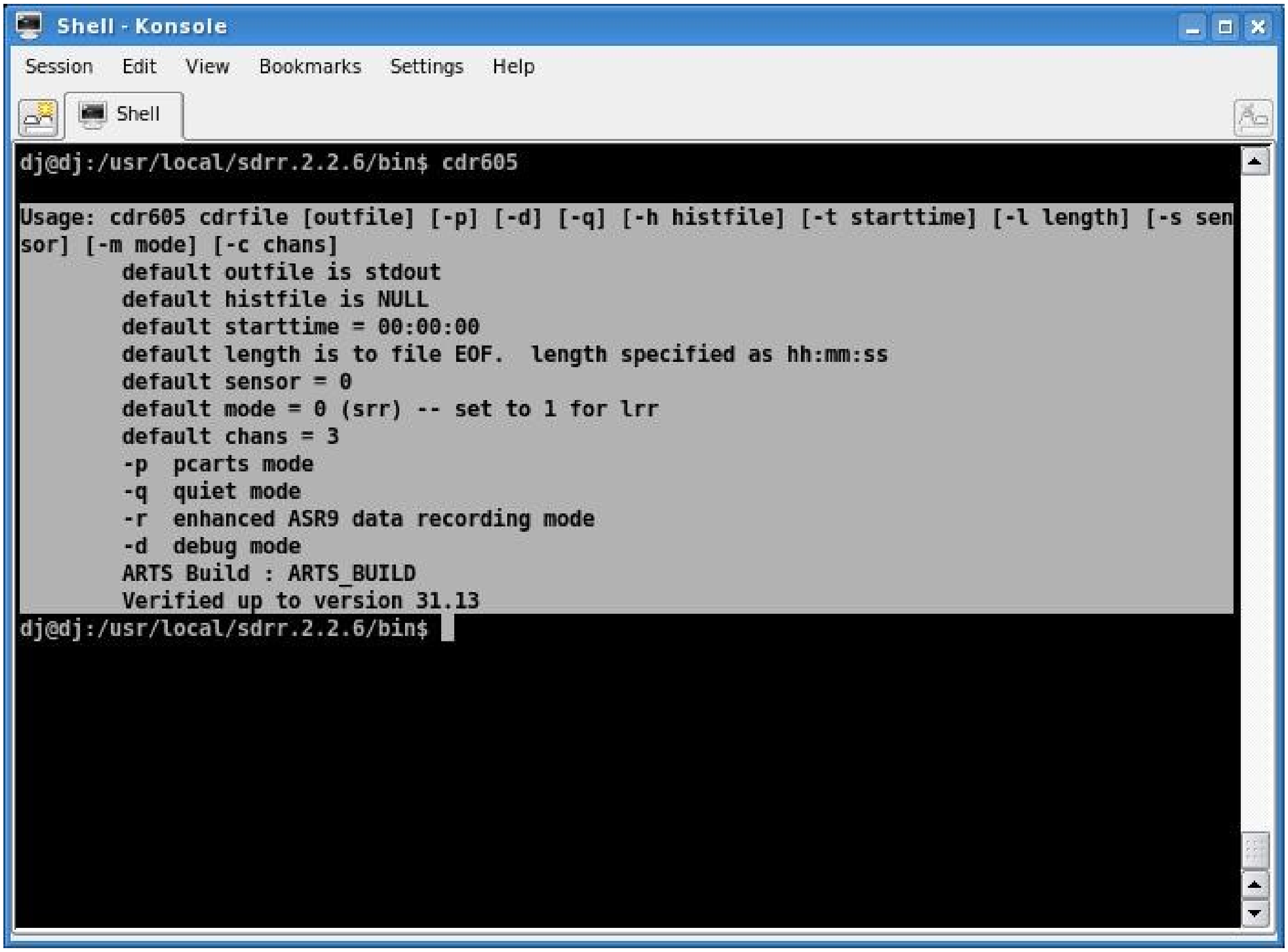
Look in: /usr/local/sdrr.2.2.6/bin/

File name:

File type: All Files (*)

7.8. cdr605, cdreA305, cdreA35, cdreA3S.2, cdreA3S.3, cdreR35.18, cdreR36.2, and if605 Utilities

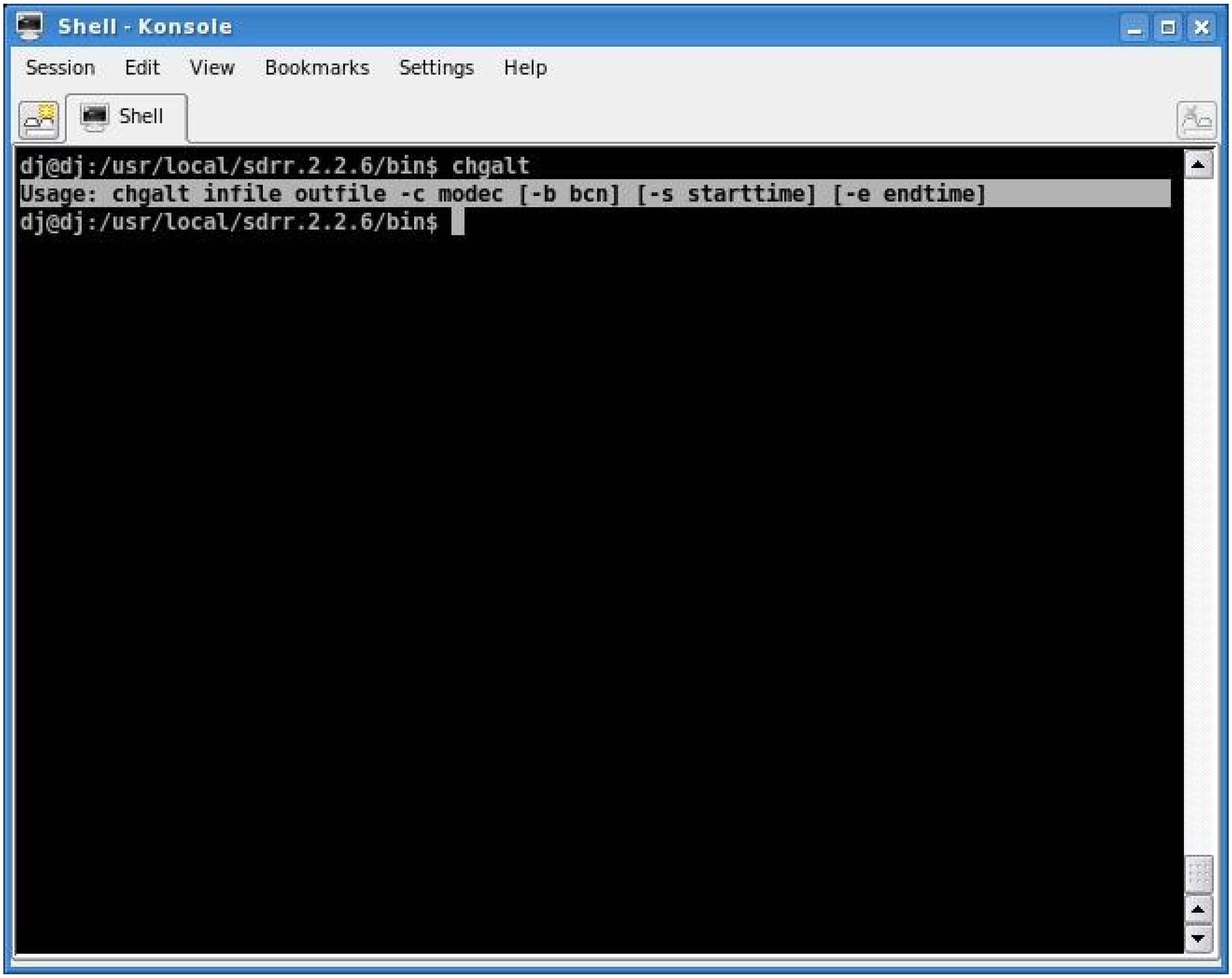
The CDR* utility suite is used for extracting CARTS radar and interfacility data. The if605 is used to extract CARTS Interfacility messages from older CDR formats. Type the utility name for a list of command options.



```
dj@dj:/usr/local/sdrr.2.2.6/bin$ cdr605

Usage: cdr605 cdrfile [outfile] [-p] [-d] [-q] [-h histfile] [-t starttime] [-l length] [-s sen
sor] [-m mode] [-c chans]
        default outfile is stdout
        default histfile is NULL
        default starttime = 00:00:00
        default length is to file EOF.  length specified as hh:mm:ss
        default sensor = 0
        default mode = 0 (srr) -- set to 1 for lrr
        default chans = 3
        -p  pcarts mode
        -q  quiet mode
        -r  enhanced ASR9 data recording mode
        -d  debug mode
        ARTS Build : ARTS_BUILD
        Verified up to version 31.13
dj@dj:/usr/local/sdrr.2.2.6/bin$
```

7.9. chgalt, chgbcn, chgchan, chgtime Utilities

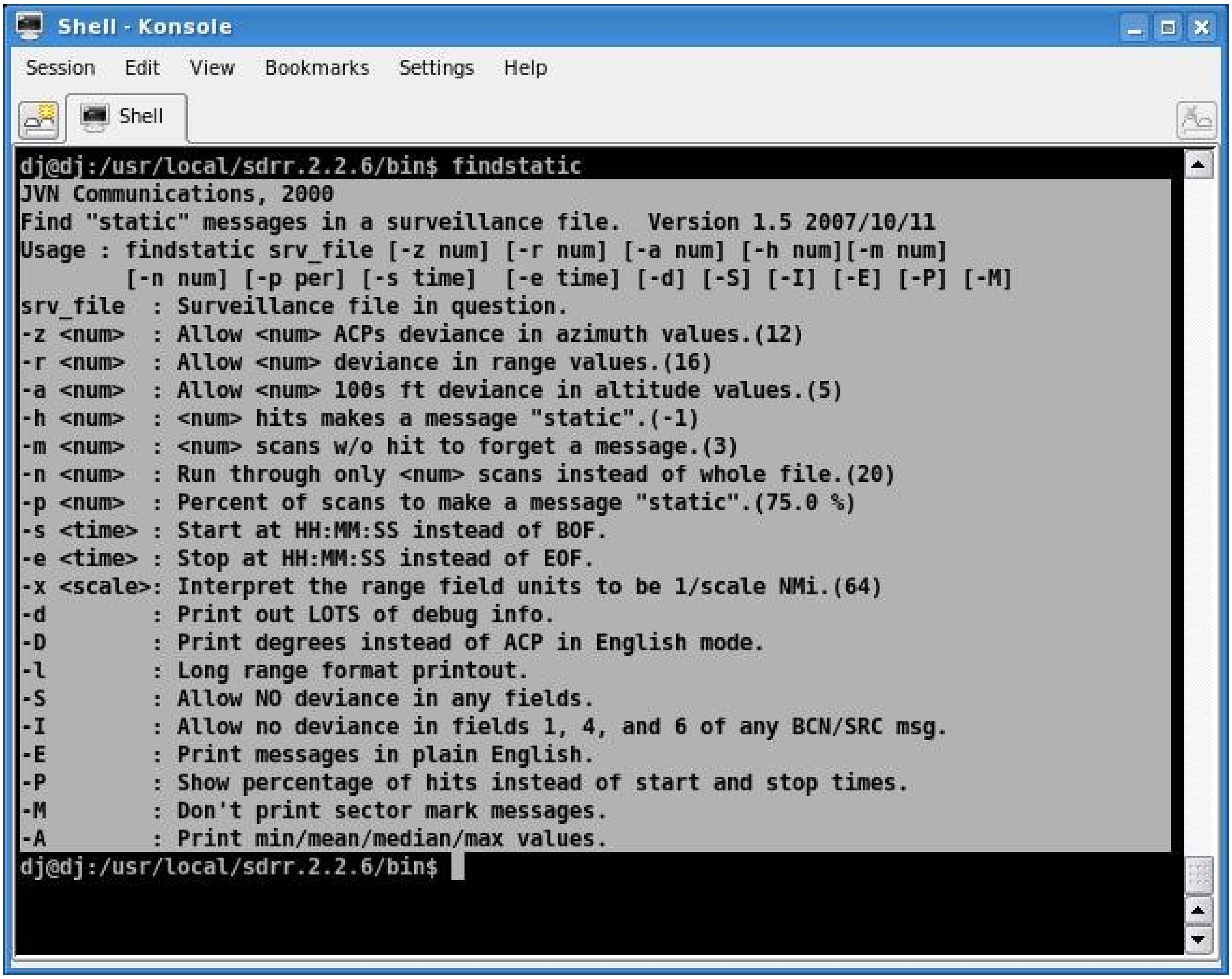
The CHG* utility suite is used to manipulate .srv files. The chgalt can be used to change a targets altitude for the duration of the .srv file. The chgbcn is used to change the beacon code, of a target, for user defined durations. The chgchan can be used to change the recorded channel number in the radar file to a user defined radar channel. The chgtime utility can be used to change the start time of an .srv file. Type the utility name for a list of command options.



```
dj@dj:/usr/local/sdrr.2.2.6/bin$ chgalt
Usage: chgalt infile outfile -c modec [-b bcn] [-s starttime] [-e endtime]
dj@dj:/usr/local/sdrr.2.2.6/bin$
```

7.10. findstatic and findtime Utilities

The find* utilities are used on .srv files as analysis tools. The findstatic utility is extremely helpful when trying to locate test targets in radar data, such as; PARROTS, PE, SRTQC, and BRTQC. The findtime can be used to see the start and end time of a .srv file. Type the utility name for a list of command options.



```
dj@dj:/usr/local/sdrr.2.2.6/bin$ findstatic
JVN Communications, 2000
Find "static" messages in a surveillance file.  Version 1.5 2007/10/11
Usage : findstatic srv_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]
srv_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.
dj@dj:/usr/local/sdrr.2.2.6/bin$
```

7.11. firs2sdrr Utility

Other, preferred utility to playback the older, FIRS scenarios is firs2sdrr. This utility converts the FIRS type scenarios to the SDRR type one.

The usage is as follows:

firs2sdrr firsfile [-o sdrrScenarioName] [-v] [-f]

-o sdrr scenario name (defaults to “sdrr”)

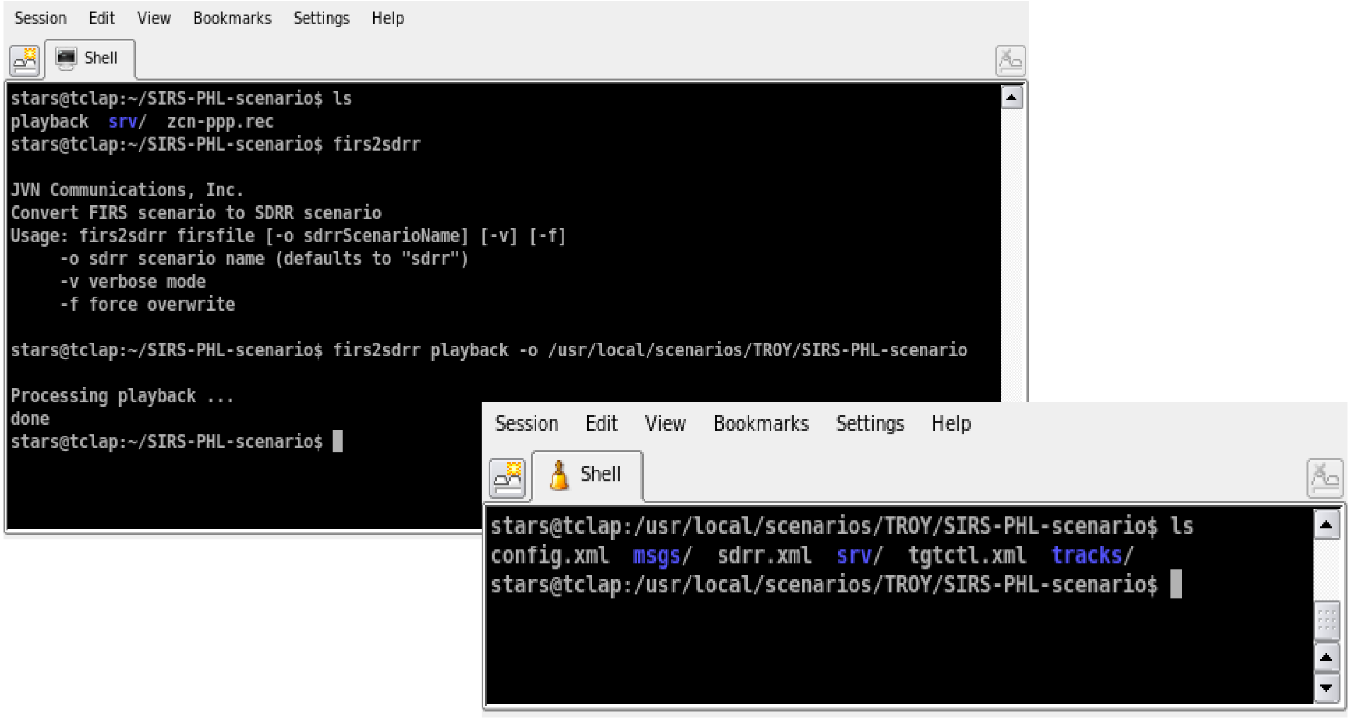
-v verbose mode

- force overwrite

*This is an example of converting the FIRS **playback** scenario located in the **/home/stars/SIRS-PHL-scenario** into a SDRR one:*

*Change to the **SIRS** directory cd **/home/stars/SIRS-PHL-scenario.***
*Run the **firs2sdrr playback –o /usr/local/scenarios/TROY/SIRS-PHL-scenario**.*
*This will convert the **SIRS** scenario to an **SDRR** scenario and place the new **SDRR** scenario in the **/usr/local/scenarios/TROY** directory.*

*Change to the **/usr/local/scenarios/TROY/SIRS-PHL-scenario** directory to see your new **SDRR** scenario.*

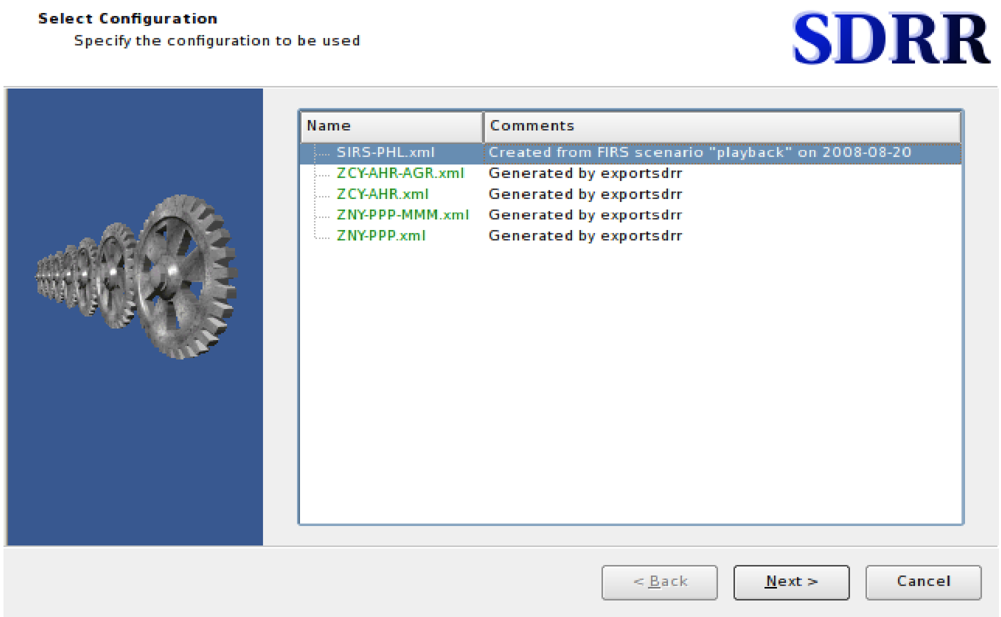


```
stars@tclap:~/SIRS-PHL-scenario$ ls
playback  srv/  zcn-ppp.rec
stars@tclap:~/SIRS-PHL-scenario$ firs2sdrr

JVN Communications, Inc.
Convert FIRS scenario to SDRR scenario
Usage: firs2sdrr firsfile [-o sdrrScenarioName] [-v] [-f]
     -o sdrr scenario name (defaults to "sdrr")
     -v verbose mode
     -f force overwrite

stars@tclap:~/SIRS-PHL-scenario$ firs2sdrr playback -o /usr/local/scenarios/TROY/SIRS-PHL-scenario

Processing playback ...
done
stars@tclap:~/SIRS-PHL-scenario$
```

```
stars@tclap:/usr/local/scenarios/TROY/SIRS-PHL-scenario$ ls
config.xml  msgs/  sdrr.xml  srv/  tgtctl.xml  tracks/
stars@tclap:/usr/local/scenarios/TROY/SIRS-PHL-scenario$
```

*The SDRR **firs2sdrr** output created configuration file located in the **/usr/local/scenarios/TROY/SIRS-PHL-scenario** directory. This file is named **config.xml**. It needs to be copied to the **/usr/local/cfg** directory. It is strongly recommended for the file to be renamed so it will not be overwritten in the next conversion and to also associate that it was created from a FIRS converted scenario (we will re-name it SIRS-PHL.xml in our example).*

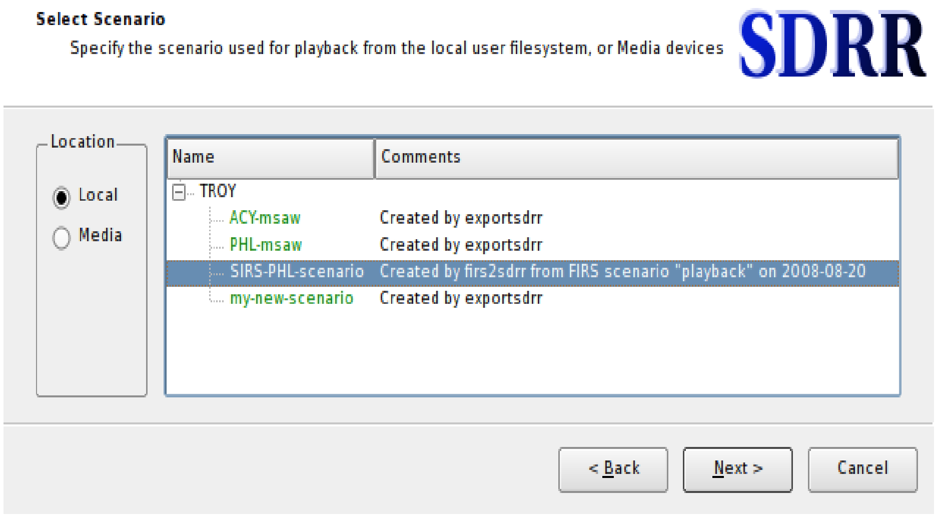
***firs2sdrr** generates additional information which is added into the **Comments** section of the scenario playback (see screenshots on the next page). This information resides in the **sdrr.xml** file and can be found in the scenario’s directory.*



Select Configuration

Specify the configuration to be used

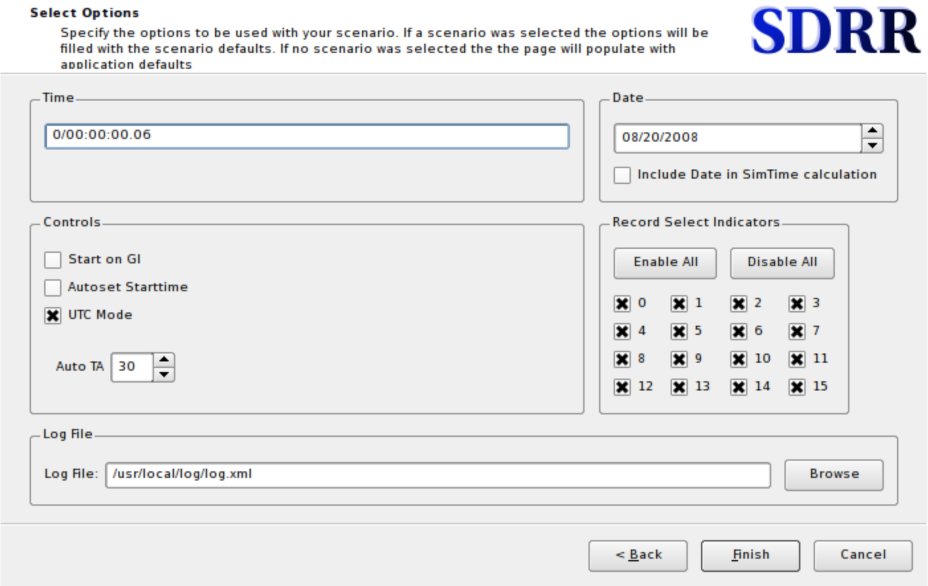
Name	Comments
SIRS-PHL.xml	Created from FIRS scenario "playback" on 2008-08-20
ZCY-AHR-AGR.xml	Generated by exportsdrr
ZCY-AHR.xml	Generated by exportsdrr
ZNY-PPP-MMM.xml	Generated by exportsdrr
ZNY-PPP.xml	Generated by exportsdrr



Select Scenario

Specify the scenario used for playback from the local user filesystem, or Media devices

Name	Comments
TROY	
ACY-msaw	Created by exportsdrr
PHL-msaw	Created by exportsdrr
SIRS-PHL-scenario	Created by firs2sdrr from FIRS scenario "playback" on 2008-08-20
my-new-scenario	Created by exportsdrr



Select Options

Specify the options to be used with your scenario. If a scenario was selected the options will be filled with the scenario defaults. If no scenario was selected the the page will populate with application defaults

Time

0/00:00:00.06

Date

08/20/2008

☐ Include Date in SimTime calculation

Controls

☐ Start on GI

☐ Autoset Starttime

☒ UTC Mode

Auto TA 30

Record Select Indicators

☒ 0 ☒ 1 ☒ 2 ☒ 3

☒ 4 ☒ 5 ☒ 6 ☒ 7

☒ 8 ☒ 9 ☒ 10 ☒ 11

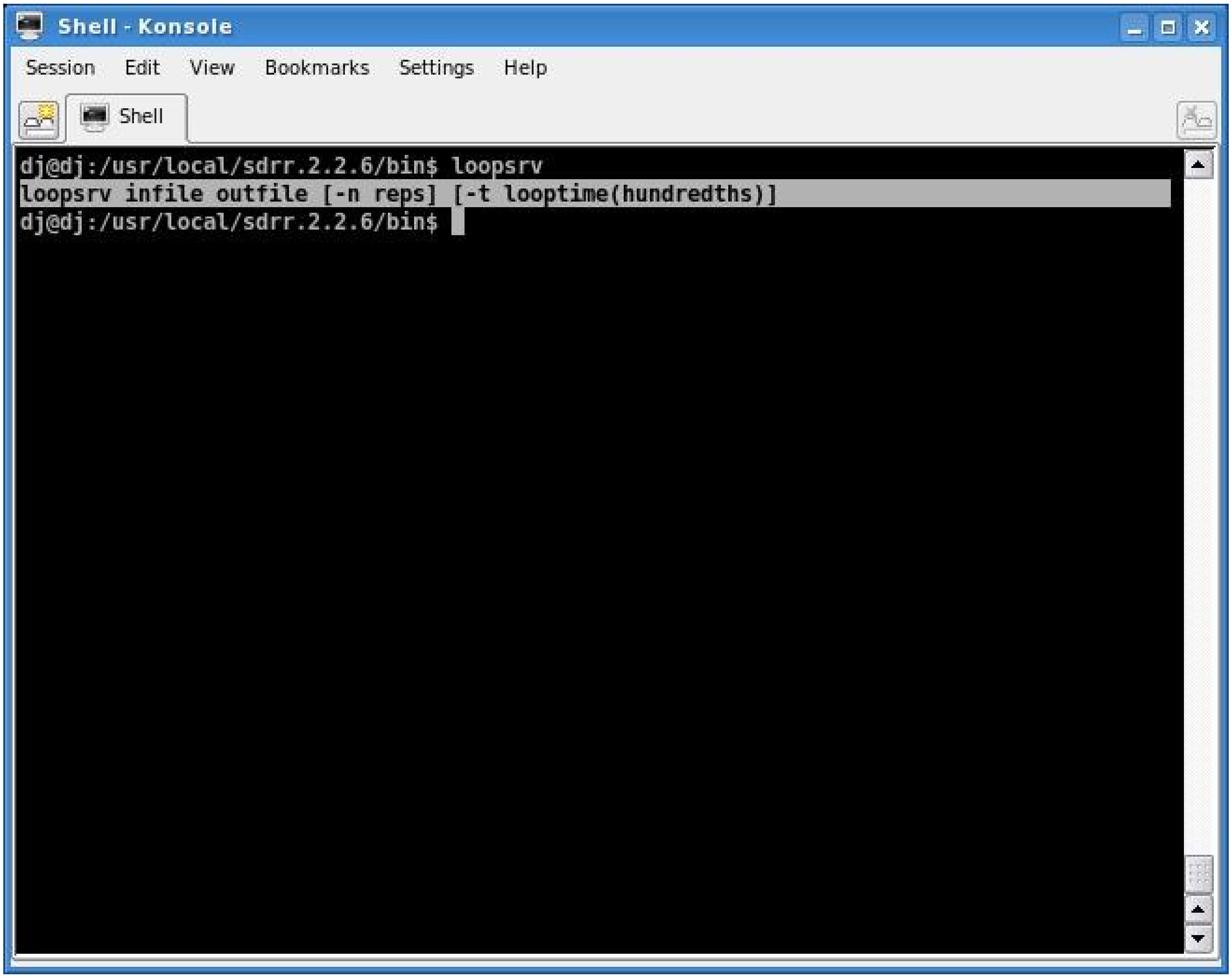
☒ 12 ☒ 13 ☒ 14 ☒ 15

Log File

Log File: /usr/local/log/log.xml

7.12. loopsrv

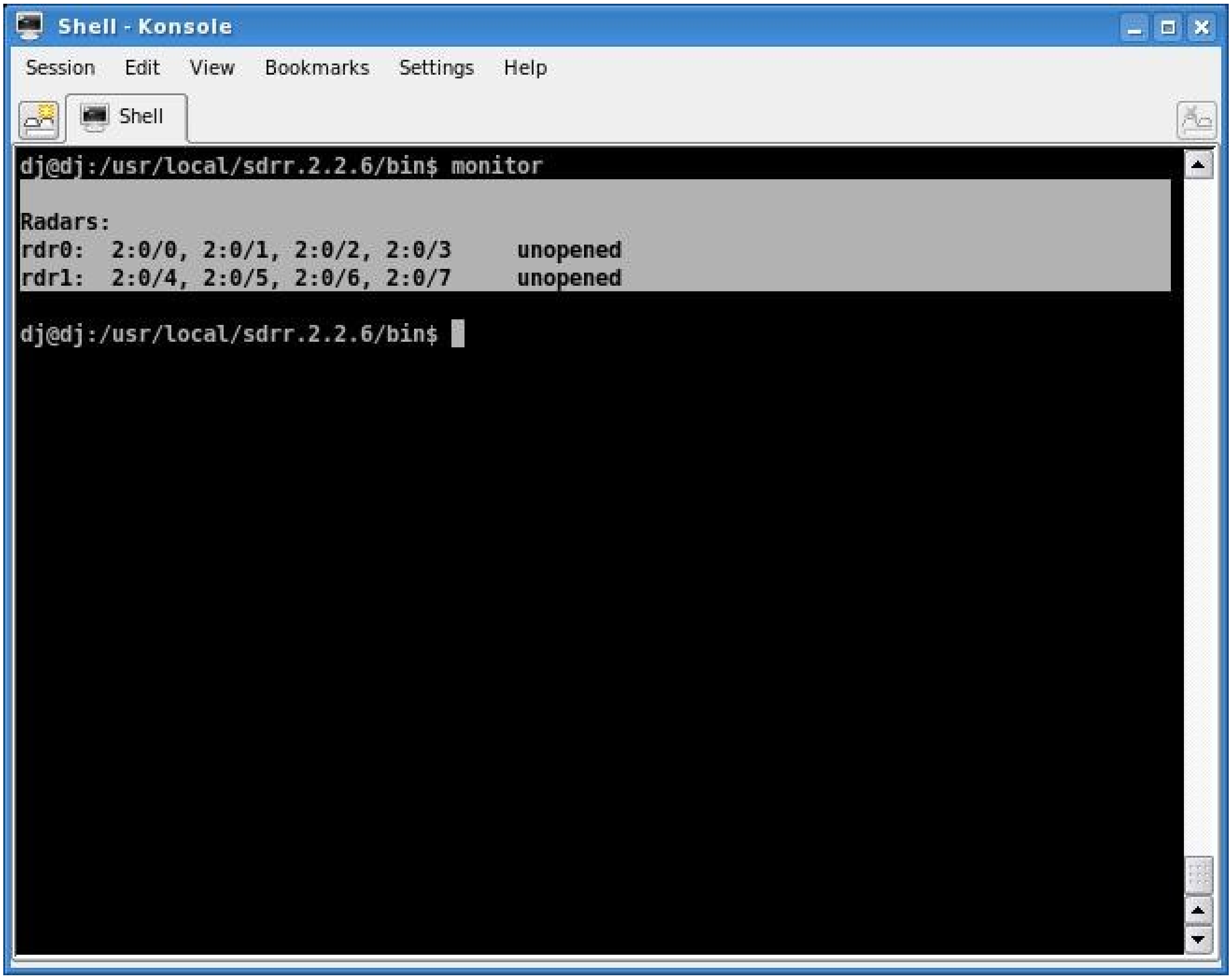
The loopsrv utility is used for extending the length of .srv files. The utility takes the original .srv file and combines itself to the end of the file, extending the play time of the file. Type the utility name for a list of command options.



```
dj@dj:/usr/local/sdrr.2.2.6/bin$ loopsrv
loopsrv infile outfile [-n reps] [-t looptime(hundredths)]
dj@dj:/usr/local/sdrr.2.2.6/bin$
```

7.13. monitor

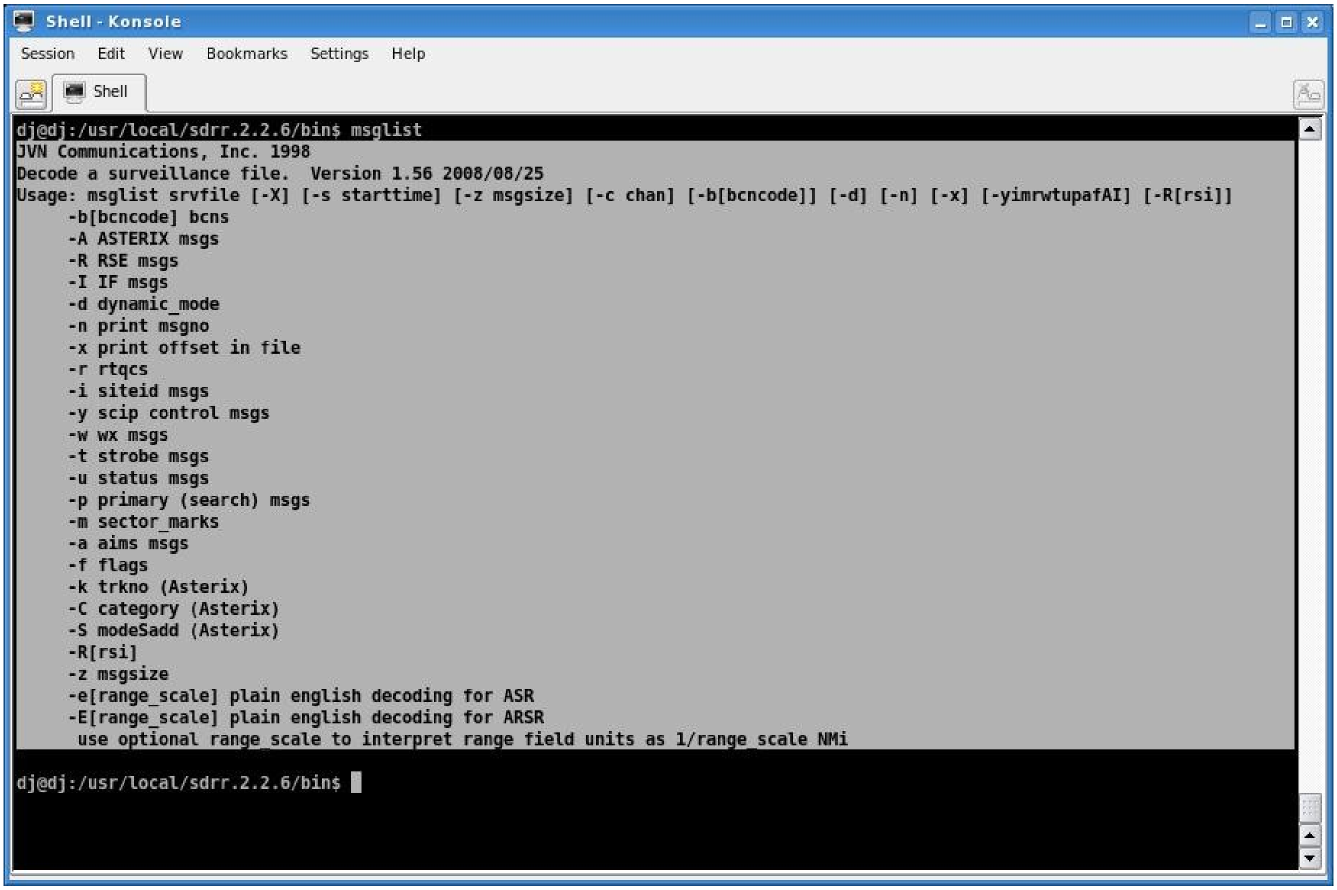
The monitor utility is used to see what radar and interfacility devices are available to the SDRR machine.



```
dj@dj:/usr/local/sdrr.2.2.6/bin$ monitor

Radars:
rdr0:  2:0/0, 2:0/1, 2:0/2, 2:0/3     unopened
rdr1:  2:0/4, 2:0/5, 2:0/6, 2:0/7     unopened

dj@dj:/usr/local/sdrr.2.2.6/bin$
```

7.14. msglist

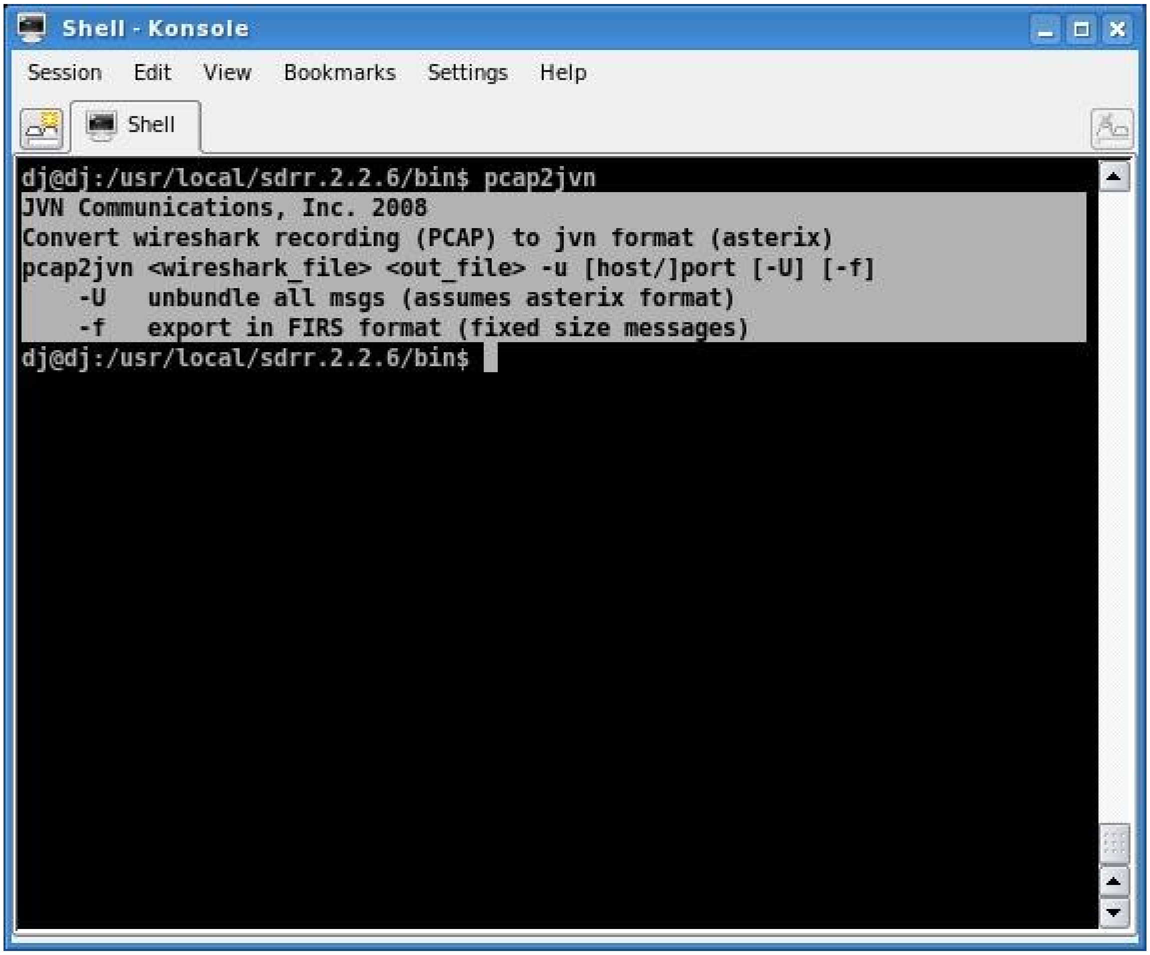
The msglist utility is one of the most commonly used utility. The msglist utility is used to view .srv radar files. The utility has many options to view the radar data with. You can view in raw radar mode, (hex), and English mode. You can search on beacon targets and many other aircraft attributes. The utility can be run on files and devices. You can run msglist on a radar device, such as ***/dev/srr0***. This is a fast way to see if valid radar data is reaching the SDRR device. Type the utility name for a list of command options.



```
dj@dj:/usr/local/sdrr.2.2.6/bin$ msglist
JVN Communications, Inc. 1998
Decode a surveillance file.  Version 1.56 2008/08/25
Usage: msglist srvfile [-X] [-s starttime] [-z msgsize] [-c chan] [-b[bcncode]] [-d] [-n] [-x] [-yimrwtupafAI] [-R[rsi]]
     -b[bcncode] bcns
     -A ASTERIX msgs
     -R RSE msgs
     -I IF msgs
     -d dynamic_mode
     -n print msgno
     -x print offset in file
     -r rtqcs
     -i siteid msgs
     -y scip control msgs
     -w wx msgs
     -t strobe msgs
     -u status msgs
     -p primary (search) msgs
     -m sector_marks
     -a aims msgs
     -f flags
     -k trkno (Asterix)
     -C category (Asterix)
     -S modeSadd (Asterix)
     -R[rsi]
     -z msgsize
     -e[range_scale] plain english decoding for ASR
     -E[range_scale] plain english decoding for ARSR
      use optional range_scale to interpret range field units as 1/range_scale NMi

dj@dj:/usr/local/sdrr.2.2.6/bin$ 
```

7.15. pcap2jvn

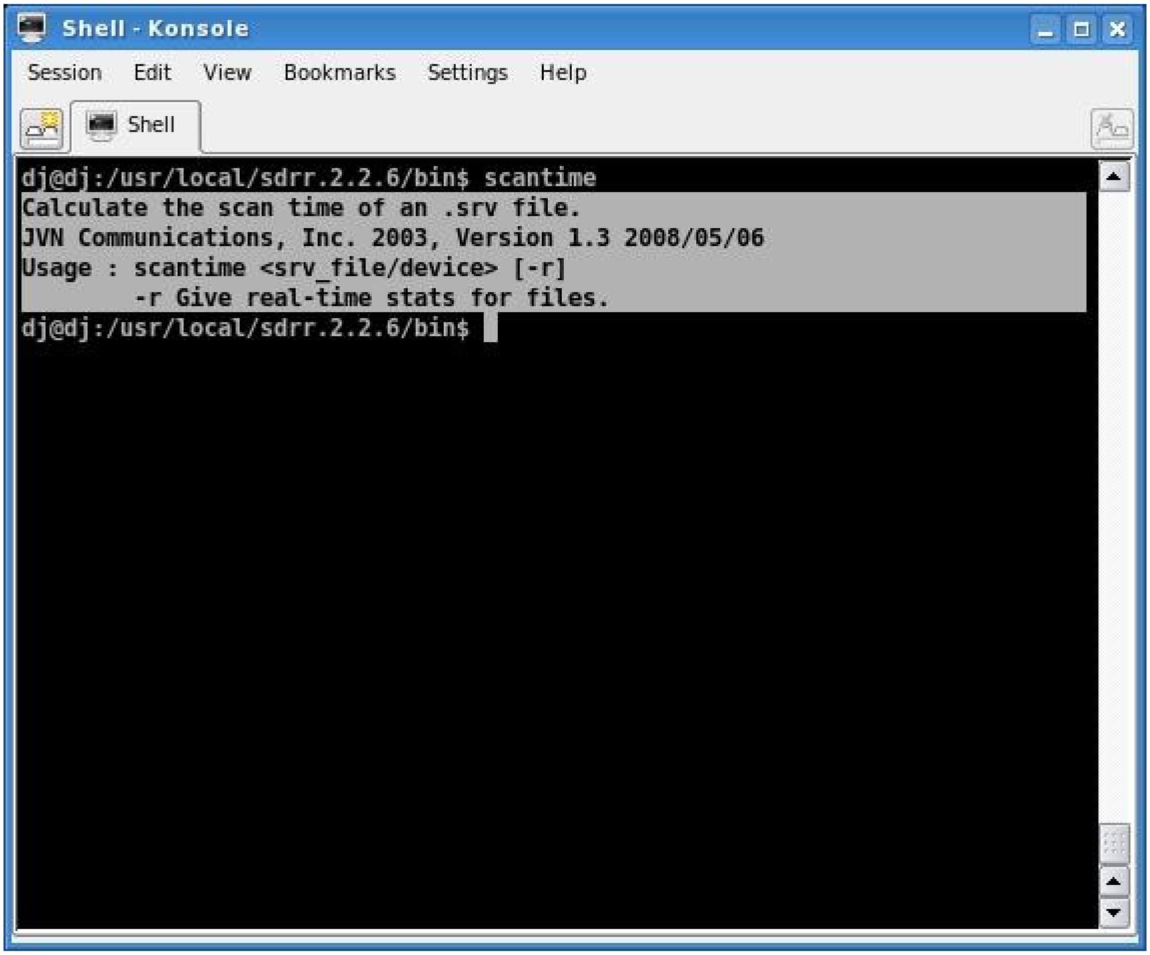
The pcap2jvn utility is used to convert a *wireshark* .cap file to SDRR .ast files. Once the files have been converted, SDRR can play them back or the user can use msglist utility view them. Type the utility name for a list of command options.



```
dj@dj:/usr/local/sdrr.2.2.6/bin$ pcap2jvn
JVN Communications, Inc. 2008
Convert wireshark recording (PCAP) to jvn format (asterix)
pcap2jvn <wireshark_file> <out_file> -u [host/]port [-U] [-f]
    -U   unbundle all msgs (assumes asterix format)
    -f   export in FIRS format (fixed size messages)
dj@dj:/usr/local/sdrr.2.2.6/bin$
```

7.16. scantime

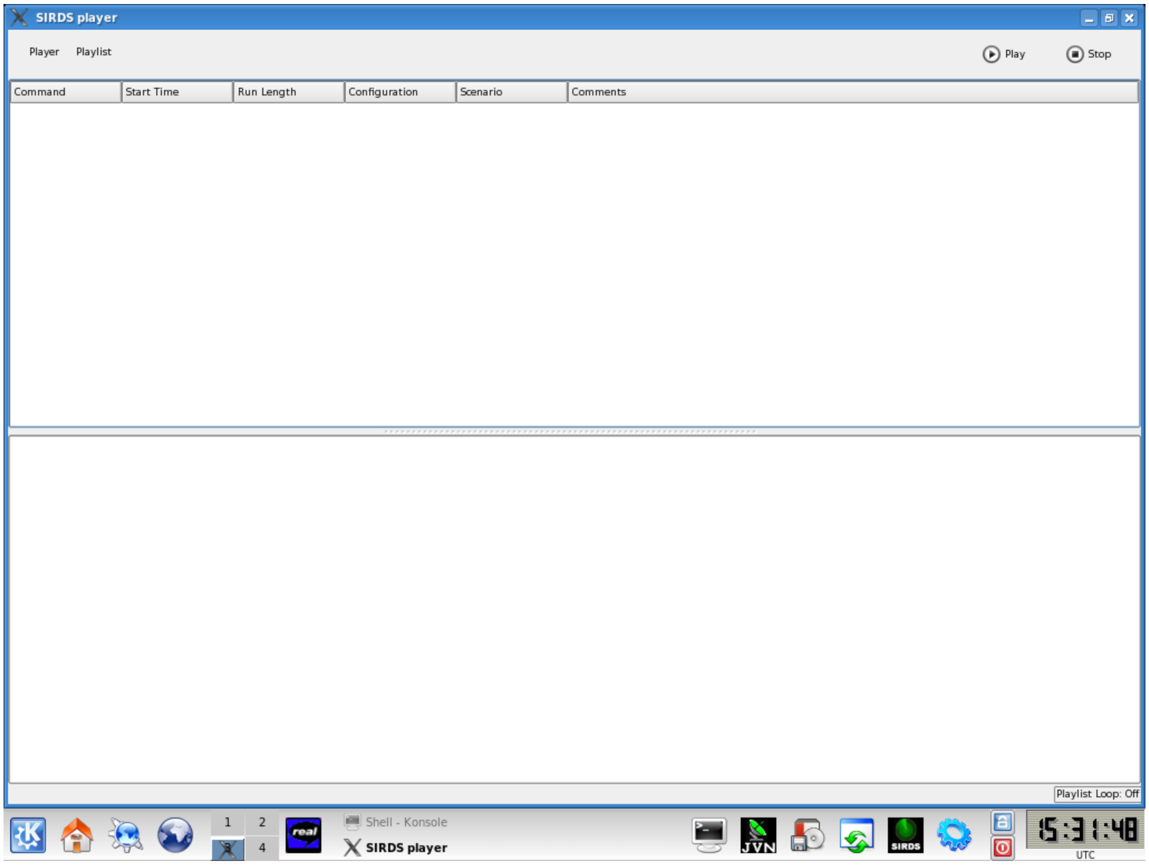
The scantime utility is used on .srv files to determine the radars scan time. Most ASR9 radar will be discovered around 4.6 seconds. Type the utility name for a list of command options.



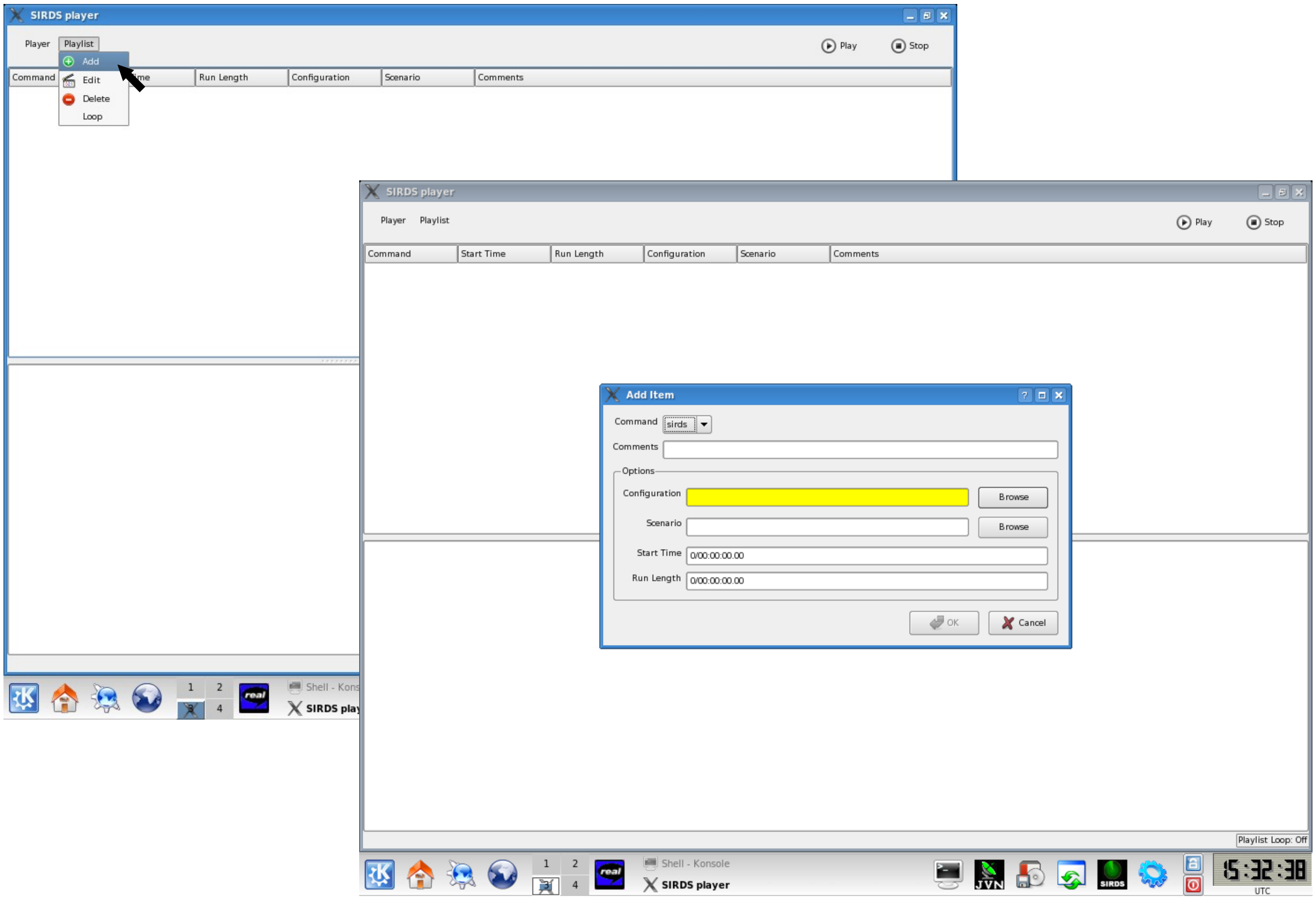
```
dj@dj:/usr/local/sdrr.2.2.6/bin$ scantime
Calculate the scan time of an .srv file.
JVN Communications, Inc. 2003, Version 1.3 2008/05/06
Usage : scantime <srv_file/device> [-r]
        -r Give real-time stats for files.
dj@dj:/usr/local/sdrr.2.2.6/bin$
```

7.17. SIRDSplayer

The SIRDSplayer utility allows users to create playlists containing SDRR scenarios. This utility can be launched either from an icon on the KDE desktop (if created) or by typing ***sirdsplayer*** (case sensitive) in a terminal window. This will launch the SIRDSplayer GUI interface:



To add a scenario to the playlist, click **Playlist** → **Add** and you will see the Add Item window:

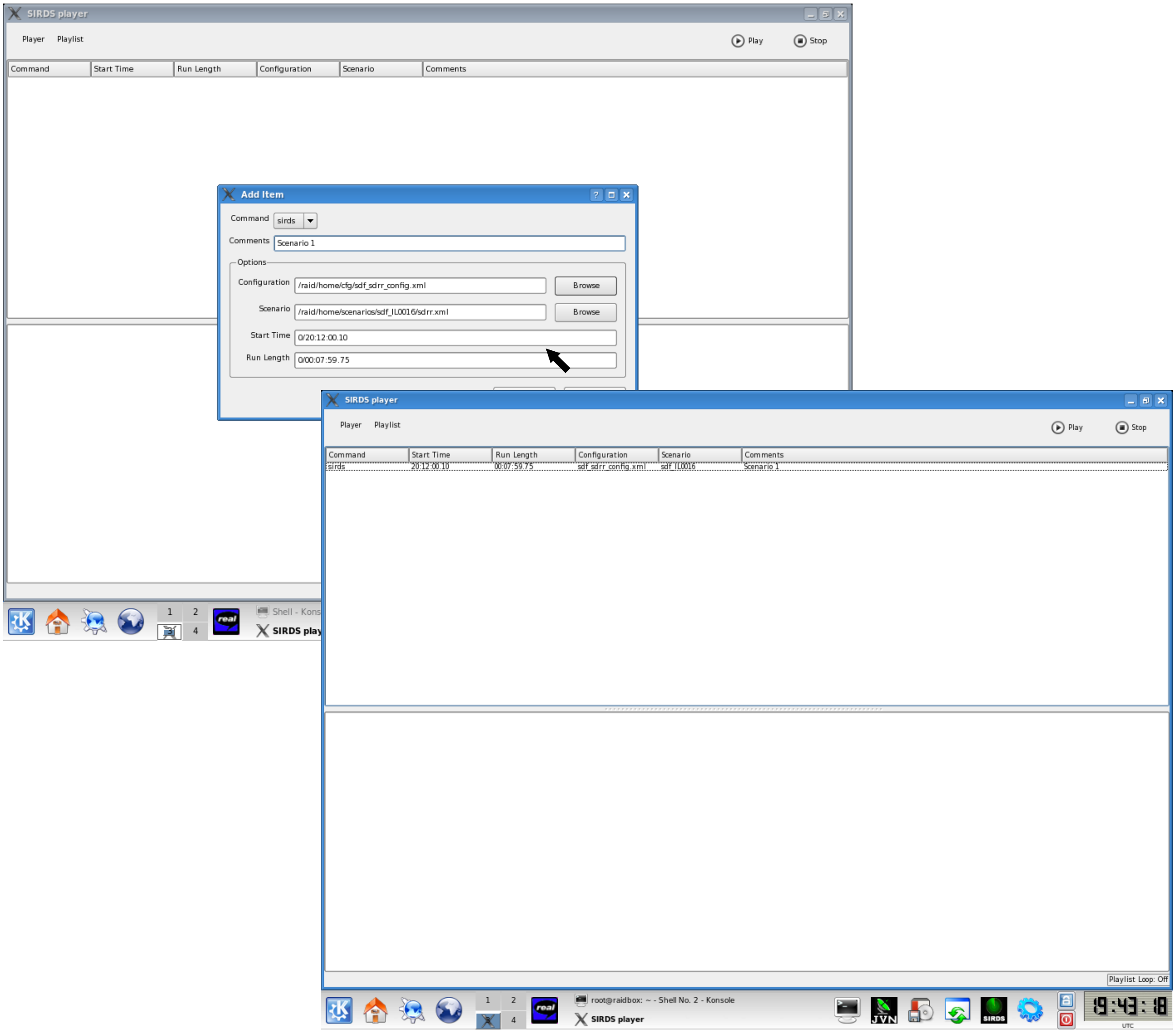


Command: sirds
Comments:
Configuration:
Scenario:
Start Time: 0/00:00:00.00
Run Length: 0/00:00:00.00

These are the available options:

Command – select to add either a sirds application or a sleep (delay) function to the playlist.
Comments – allows user to add a comment for each item in the playlist.
Configuration – enter the full path or browse to select the configuration file.
Scenario – enter the full path or browse to select the scenario file.
Start Time – this field will automatically populate when the scenario is selected, however user can adjust the start time to that of interest.
Run Length – this field will automatically populate when the scenario is selected, however user can adjust the run length to narrow down the time of interest.

Once Ok is clicked, the item is added to the playlist:



Command: sirds

Comments: Scenario 1

Configuration: /raid/home/cfg/sdf_sdrr_config.xml

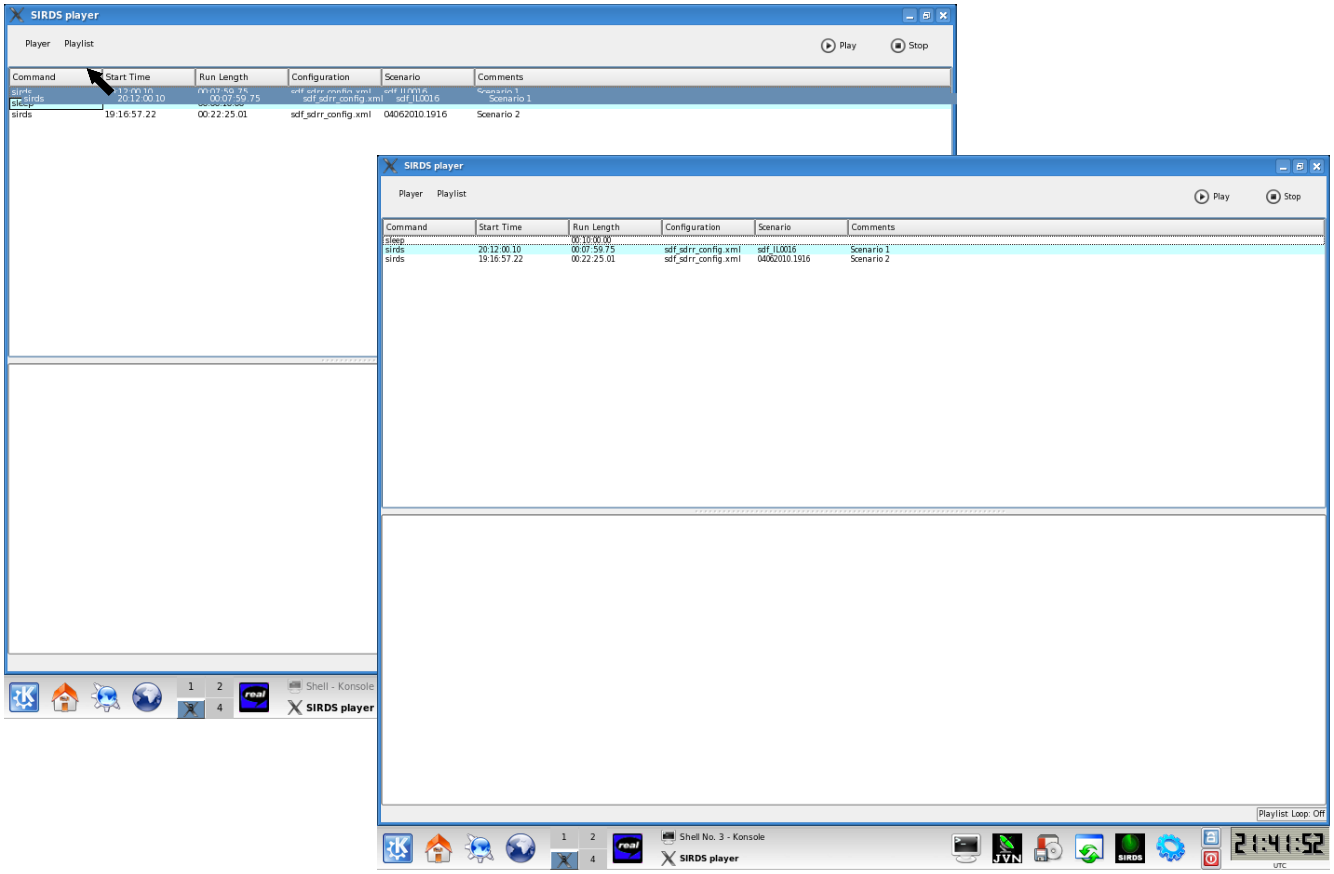
Scenario: /raid/home/scenarios/sdf_IL0016/sdrr.xml

Start Time: 0/20:12:00.10

Run Length: 0/00:07:59.75

Command	Start Time	Run Length	Configuration	Scenario	Comments
sirds	20:12:00.10	00:07:59.75	sdf_sdrr_config.xml	sdf_IL0016	Scenario 1

From the playlist window, you can drag and drop the items to manipulate the order:

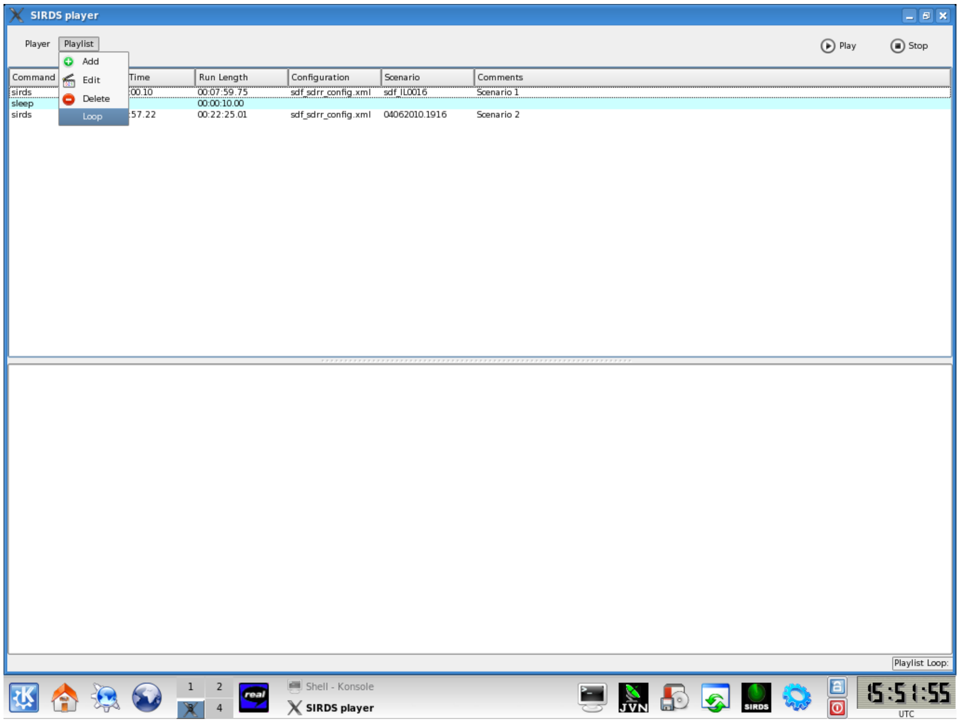


Command	Start Time	Run Length	Configuration	Scenario	Comments
sirds	20:12:00.10	00:07:59.75	sdf_sdrr_config.xml	sdf_IL0016	Scenario 1
sleep					
sirds	19:16:57.22	00:22:25.01	sdf_sdrr_config.xml	04062010.1916	Scenario 2

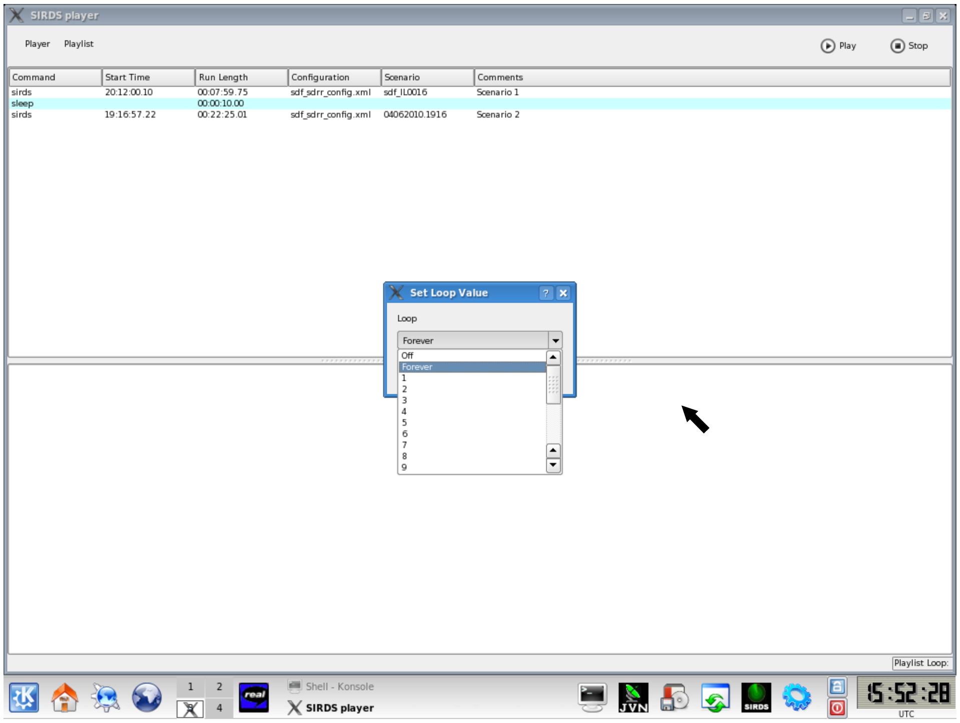
Command	Start Time	Run Length	Configuration	Scenario	Comments
sleep		00:10:00.00			
sirds	20:12:00.10	00:07:59.75	sdf_sdrr_config.xml	sdf_IL0016	Scenario 1
sirds	19:16:57.22	00:22:25.01	sdf_sdrr_config.xml	04062010.1916	Scenario 2

Likewise, to preform a copy, hold down the Ctrl key while dragging and dropping the item (you will see a plus sign with a box around it):

Once you have all your items in the playlist, click the play button to play through the playlist once or click **Playlist** → **Loop** to select the amount of times you want the playlist to loop.

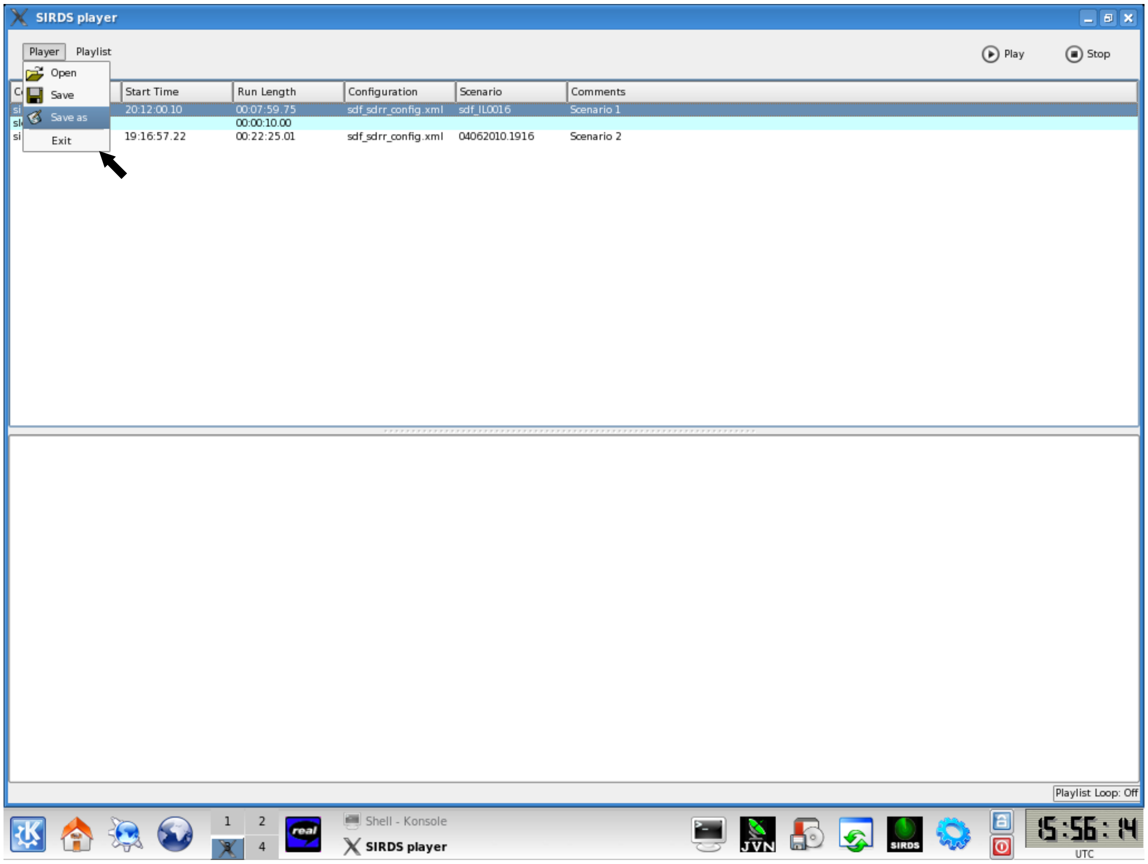


Command	Time	Run Length	Configuration	Scenario	Comments
sirds	:00.10	00:07:59.75	sdf_sdrr_config.xml	sdf_IL0016	Scenario 1
sleep		00:00:10.00			
sirds	:57.22	00:22:25.01	sdf_sdrr_config.xml	04062010.1916	Scenario 2



Command	Start Time	Run Length	Configuration	Scenario	Comments
sirds	20:12:00.10	00:07:59.75	sdf_sdrr_config.xml	sdf_IL0016	Scenario 1
sleep		00:00:10.00			
sirds	19:16:57.22	00:22:25.01	sdf_sdrr_config.xml	04062010.1916	Scenario 2

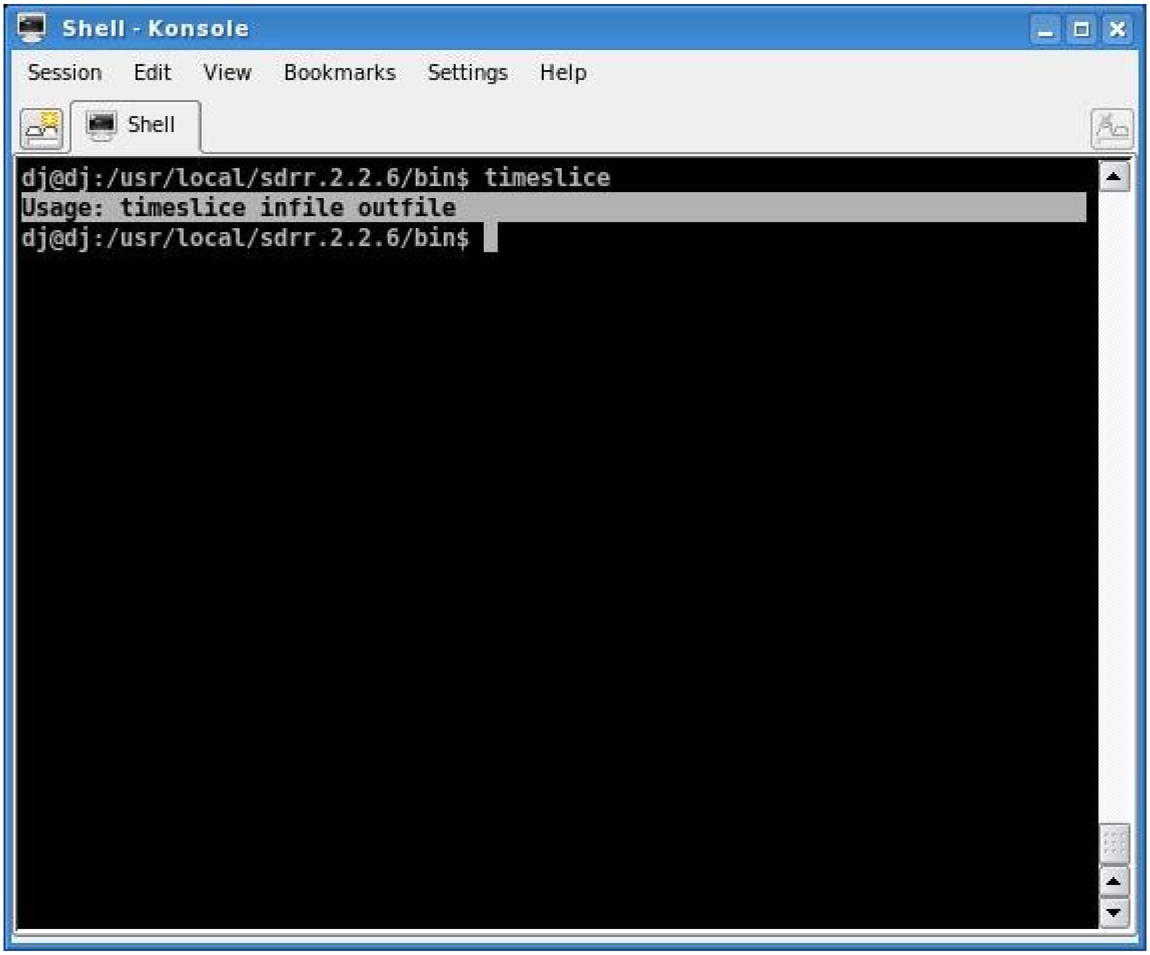
You can save the playlist by clicking **Player** → **Save As**:



C	Start Time	Run Length	Configuration	Scenario	Comments
si	20:12:00.10	00:07:59.75	sdf_sdrr_config.xml	sdf_IL0016	Scenario 1
sl		00:00:10.00			
si	19:16:57.22	00:22:25.01	sdf_sdrr_config.xml	04062010.1916	Scenario 2

7.18. timeslice

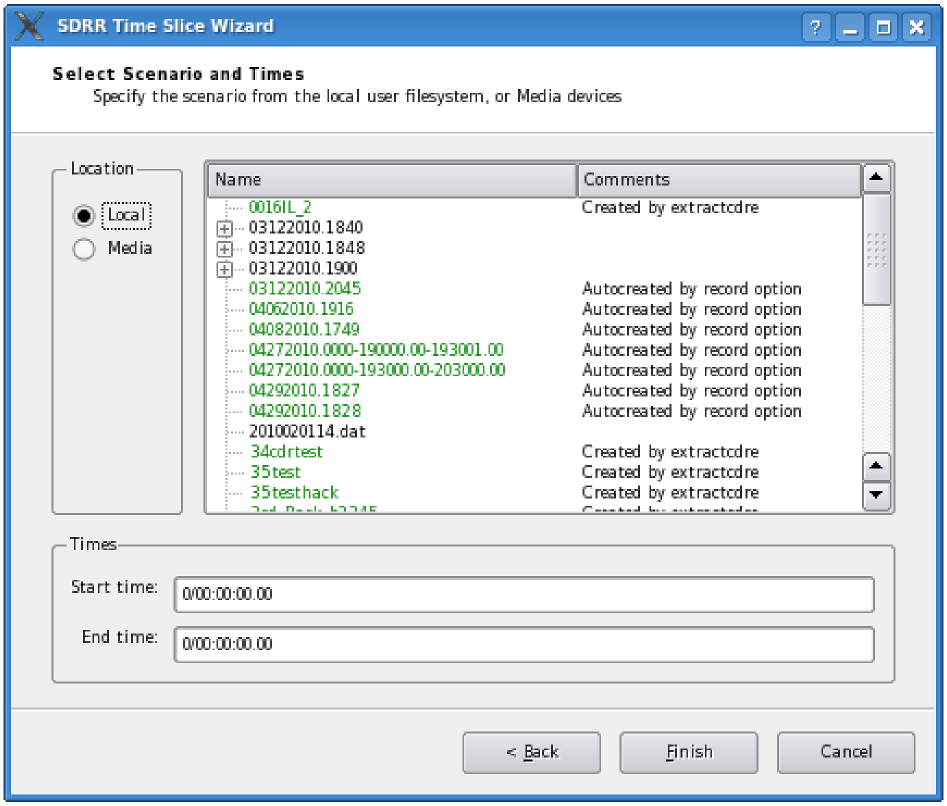
The time slice utility is used to pull a section of radar data from a larger file. If you have a ten hour scenario and only need the first thirty minutes, you can use timeslice to cut out and rename the first thirty minutes. Type the utility name for a list of command options.



```
dj@dj:/usr/local/sdrr.2.2.6/bin$ timeslice
Usage: timeslice infile outfile
dj@dj:/usr/local/sdrr.2.2.6/bin$
```

7.19. TimeSlice GUI

The TimeSlice utility 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 KDE desktop (if created) or by typing ***TimeSlice*** (case sensitive) in a terminal window. This will launch the TimeSlice GUI interface:



Select Scenario and Times

Specify the scenario from the local user filesystem, or Media devices

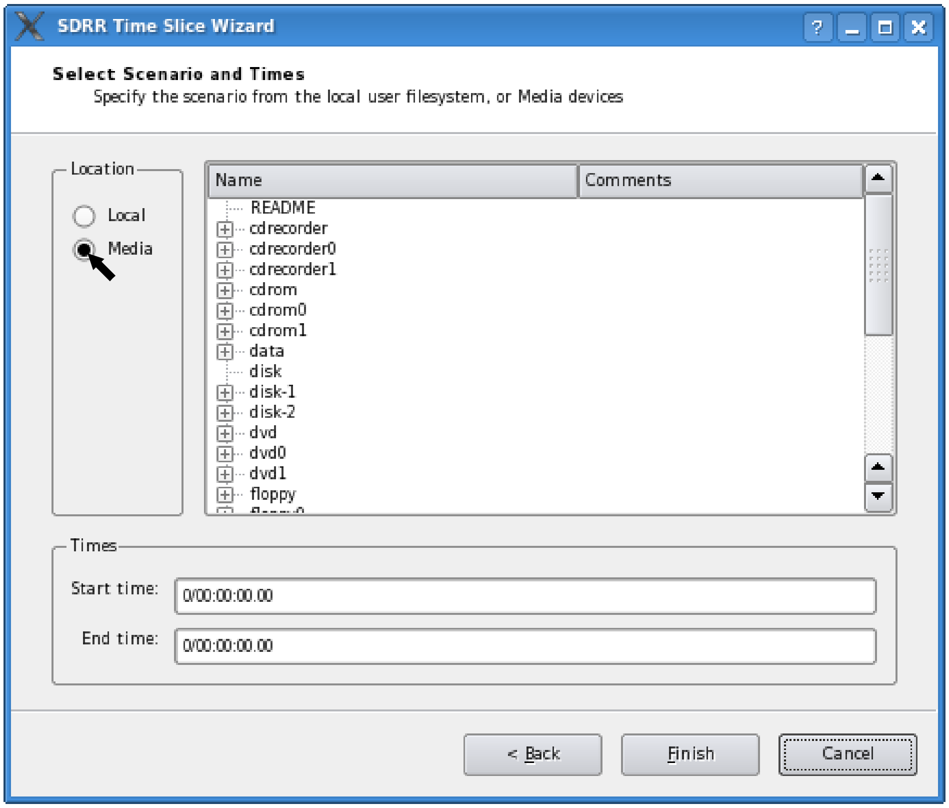
Name	Comments
0016IL_2	Created by extractcdre
03122010.1840	
03122010.1848	
03122010.1900	
03122010.2045	Autocreated by record option
04062010.1916	Autocreated by record option
04082010.1749	Autocreated by record option
04272010.0000-190000.00-193001.00	Autocreated by record option
04272010.0000-193000.00-203000.00	Autocreated by record option
04292010.1827	Autocreated by record option
04292010.1828	Autocreated by record option
2010020114.dat	
34cdrtest	Created by extractcdre
35test	Created by extractcdre
35testhack	Created by extractcdre

Times

Start time: 0/00:00:00.00

End time: 0/00:00:00.00

When the utility is launched, it defaults to the local user filesystem (/usr/local/scenarios), but the user can also select a scenario from a media device:



Select Scenario and Times

Specify the scenario from the local user filesystem, or Media devices

Location

○ Local

● Media

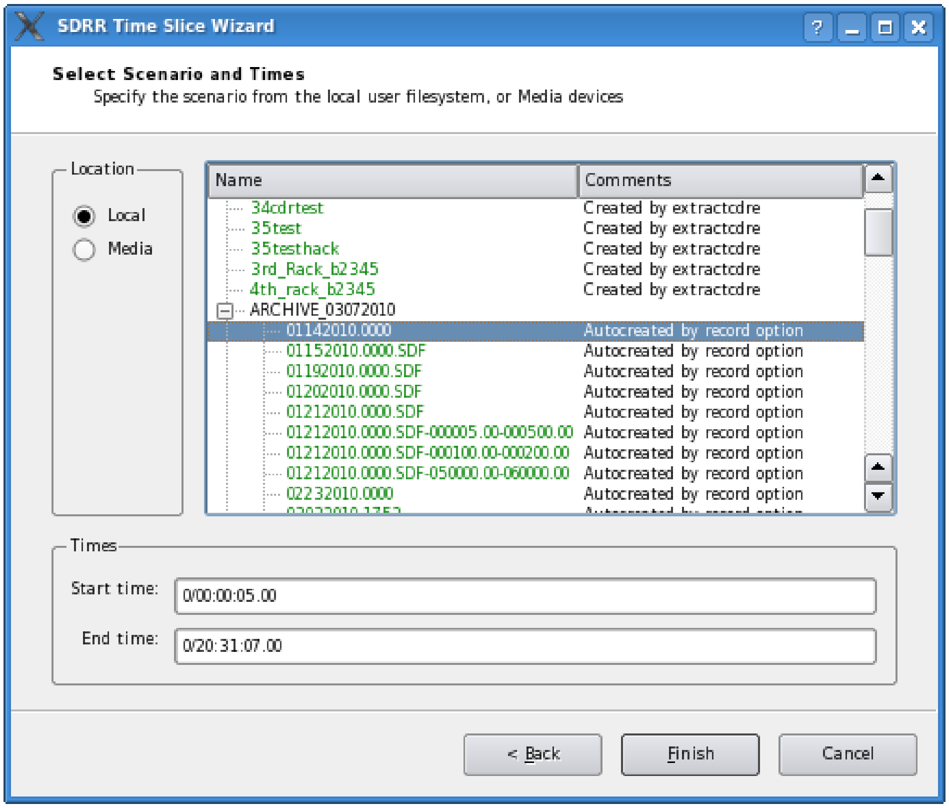
Name	Comments
README	
⊞ cdrecorder	
⊞ cdrecorder0	
⊞ cdrecorder1	
⊞ cdrom	
⊞ cdrom0	
⊞ cdrom1	
⊞ data	
disk	
⊞ disk-1	
⊞ disk-2	
⊞ dvd	
⊞ dvd0	
⊞ dvd1	
⊞ floppy	

Times

Start time: 0/00:00:00.00

End time: 0/00:00:00.00

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



Select Scenario and Times

Specify the scenario from the local user filesystem, or Media devices

Location

● Local

○ Media

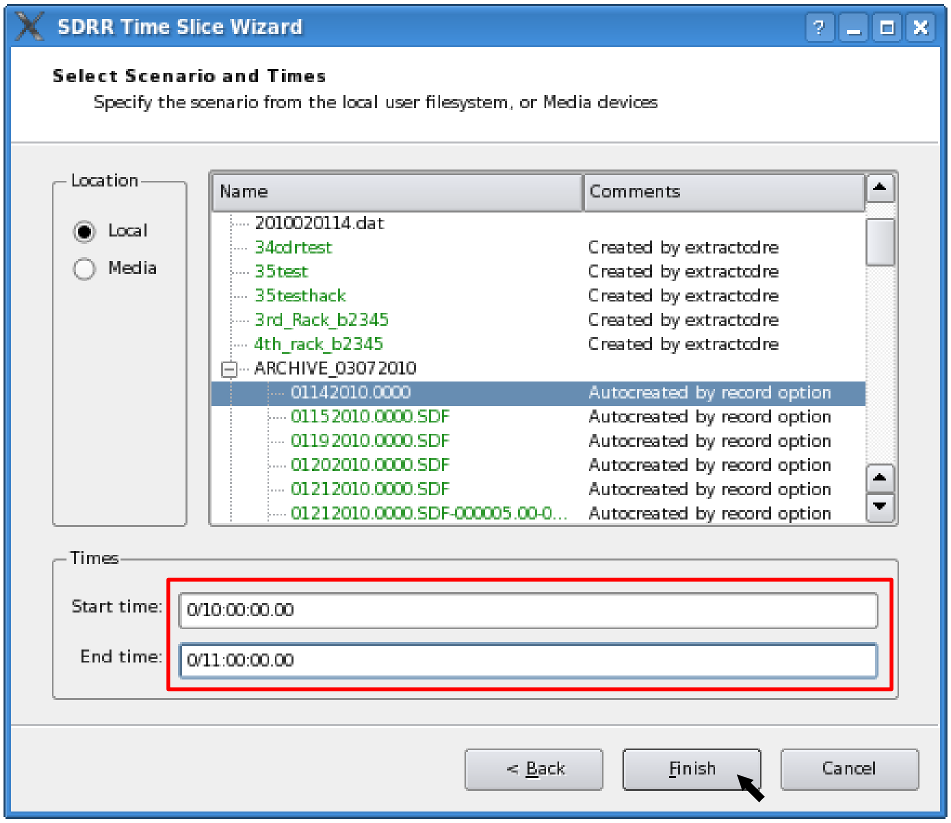
Name	Comments
34cdrtest	Created by extractcdre
35test	Created by extractcdre
35testhack	Created by extractcdre
3rd_Rack_b2345	Created by extractcdre
4th_rack_b2345	Created by extractcdre
⊟ ARCHIVE_03072010	
01142010.0000	Autocreated by record option
01152010.0000.SDF	Autocreated by record option
01192010.0000.SDF	Autocreated by record option
01202010.0000.SDF	Autocreated by record option
01212010.0000.SDF	Autocreated by record option
01212010.0000.SDF-000005.00-000500.00	Autocreated by record option
01212010.0000.SDF-000100.00-000200.00	Autocreated by record option
01212010.0000.SDF-050000.00-060000.00	Autocreated by record option
02232010.0000	Autocreated by record option

Times

Start time: 0/00:00:05.00

End time: 0/20:31:07.00

Change the start time and/or end time to narrow the scenario down to the desired time. For example, if the event of interest occurred between 10:00:00:00 and 11:00:00:00, enter 0/10:00:00:00 in the start time box and 0/11:00:00:00 in the end time box and click Finish:



SDRR Time Slice Wizard

Select Scenario and Times

Specify the scenario from the local user filesystem, or Media devices

Location

● Local

○ Media

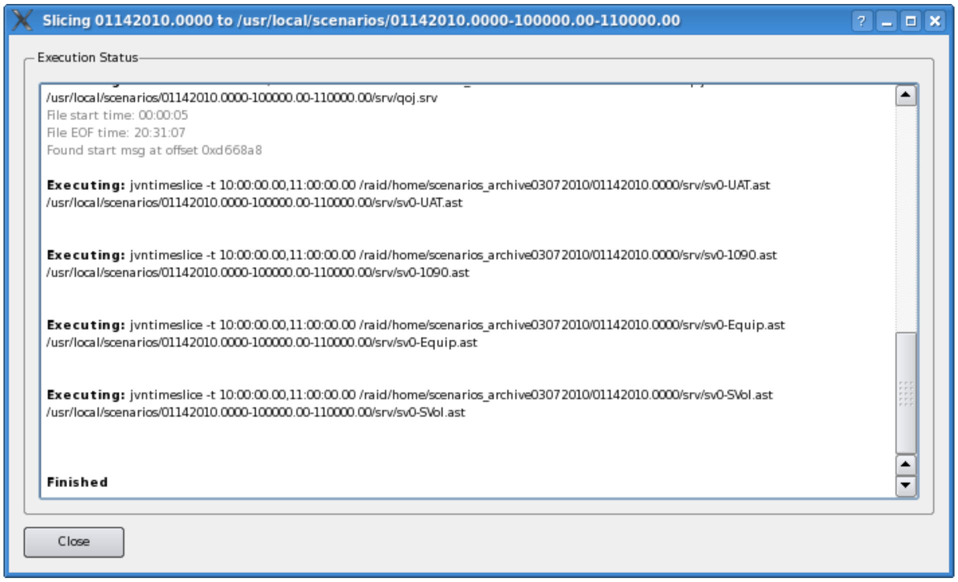
Name	Comments
2010020114.dat	
34cdrtest	Created by extractcdre
35test	Created by extractcdre
35testhack	Created by extractcdre
3rd_Rack_b2345	Created by extractcdre
4th_rack_b2345	Created by extractcdre
ARCHIVE_03072010	
01142010.0000	Autocreated by record option
01152010.0000.SDF	Autocreated by record option
01192010.0000.SDF	Autocreated by record option
01202010.0000.SDF	Autocreated by record option
01212010.0000.SDF	Autocreated by record option
01212010.0000.SDF-000005.00-0...	Autocreated by record option

Times

Start time: 0/10:00:00.00

End time: 0/11:00:00.00

You will then see the execution status window showing the individual steps that are executed in order to slice the scenario. When it is done executing it will say Finished at the bottom of the status window:

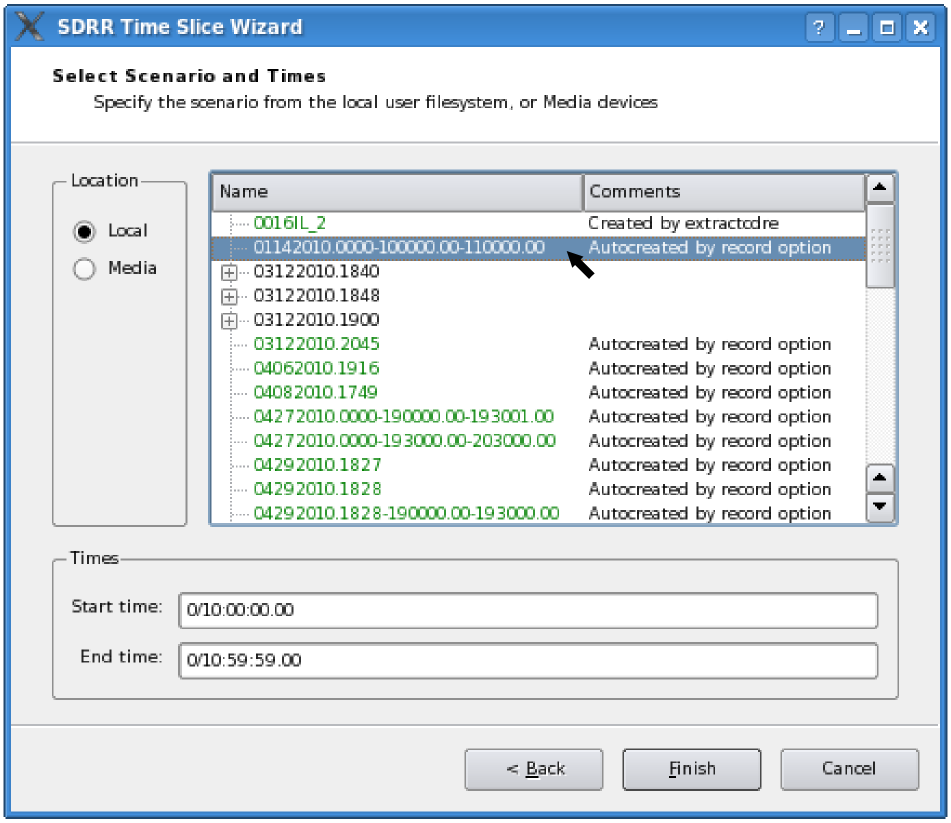


Slicing 01142010.0000 to /usr/local/scenarios/01142010.0000-100000.00-110000.00

Execution Status

```
/usr/local/scenarios/01142010.0000-100000.00-110000.00/srv/qoj.srv
File start time: 00:00:05
File EOF time: 20:31:07
Found start msg at offset 0xd668a8

Executing: jvntimeslice -t 10:00:00.00,11:00:00.00 /raid/home/scenarios_archive03072010/01142010.0000/srv/sv0-UAT.ast
/usr/local/scenarios/01142010.0000-100000.00-110000.00/srv/sv0-UAT.ast


Executing: jvntimeslice -t 10:00:00.00,11:00:00.00 /raid/home/scenarios_archive03072010/01142010.0000/srv/sv0-1090.ast
/usr/local/scenarios/01142010.0000-100000.00-110000.00/srv/sv0-1090.ast


Executing: jvntimeslice -t 10:00:00.00,11:00:00.00 /raid/home/scenarios_archive03072010/01142010.0000/srv/sv0-Equip.ast
/usr/local/scenarios/01142010.0000-100000.00-110000.00/srv/sv0-Equip.ast


Executing: jvntimeslice -t 10:00:00.00,11:00:00.00 /raid/home/scenarios_archive03072010/01142010.0000/srv/sv0-SVol.ast
/usr/local/scenarios/01142010.0000-100000.00-110000.00/srv/sv0-SVol.ast



Finished
```

Click close, and the time slice is now complete. You can check to see that the new slice was saved by opening a terminal and going to the /usr/local/scenarios/ directory, or by opening the TimeSlice GUI again:



Select Scenario and Times

Specify the scenario from the local user filesystem, or Media devices

Location

● Local

○ Media

Name	Comments
0016IL_2	Created by extractcdre
01142010.0000-100000.00-110000.00	Autocreated by record option
⊞ 03122010.1840	
⊞ 03122010.1848	
⊞ 03122010.1900	
03122010.2045	Autocreated by record option
04062010.1916	Autocreated by record option
04082010.1749	Autocreated by record option
04272010.0000-190000.00-193001.00	Autocreated by record option
04272010.0000-193000.00-203000.00	Autocreated by record option
04292010.1827	Autocreated by record option
04292010.1828	Autocreated by record option
04292010.1828-190000.00-193000.00	Autocreated by record option

Times

Start time: 0/10:00:00.00

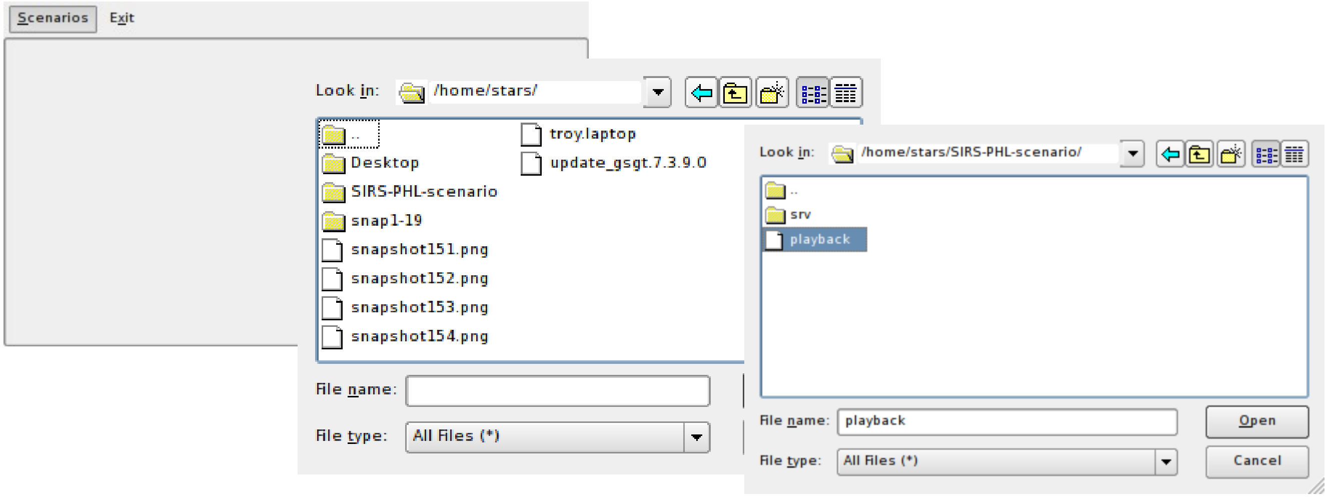
End time: 0/10:59:59.00

The new slice is saved as the original filename with the start time and end time appended to the end.

7.20. xtest Utility

The **xtest** utility is an interim solution to support legacy FIRS scenarios until all SDRR users convert their FIRS scenarios to SDRR using the firs2sdrr utility. This is strongly advised due to the fact that xtest will not be supported in the future SDRR software releases.

This utility can be launched either from an icon on the KDE desktop (if created) or by typing **xtest** in a terminal window. This will launch the XTEST GUI interface. Clicking the **Scenarios** button will open the file manager.



Look in: /home/stars/

..
Desktop
SIRS-PHL-scenario
snap1-19
snapshot151.png
snapshot152.png
snapshot153.png
snapshot154.png
troy.laptop
update_gsgt.7.3.9.0

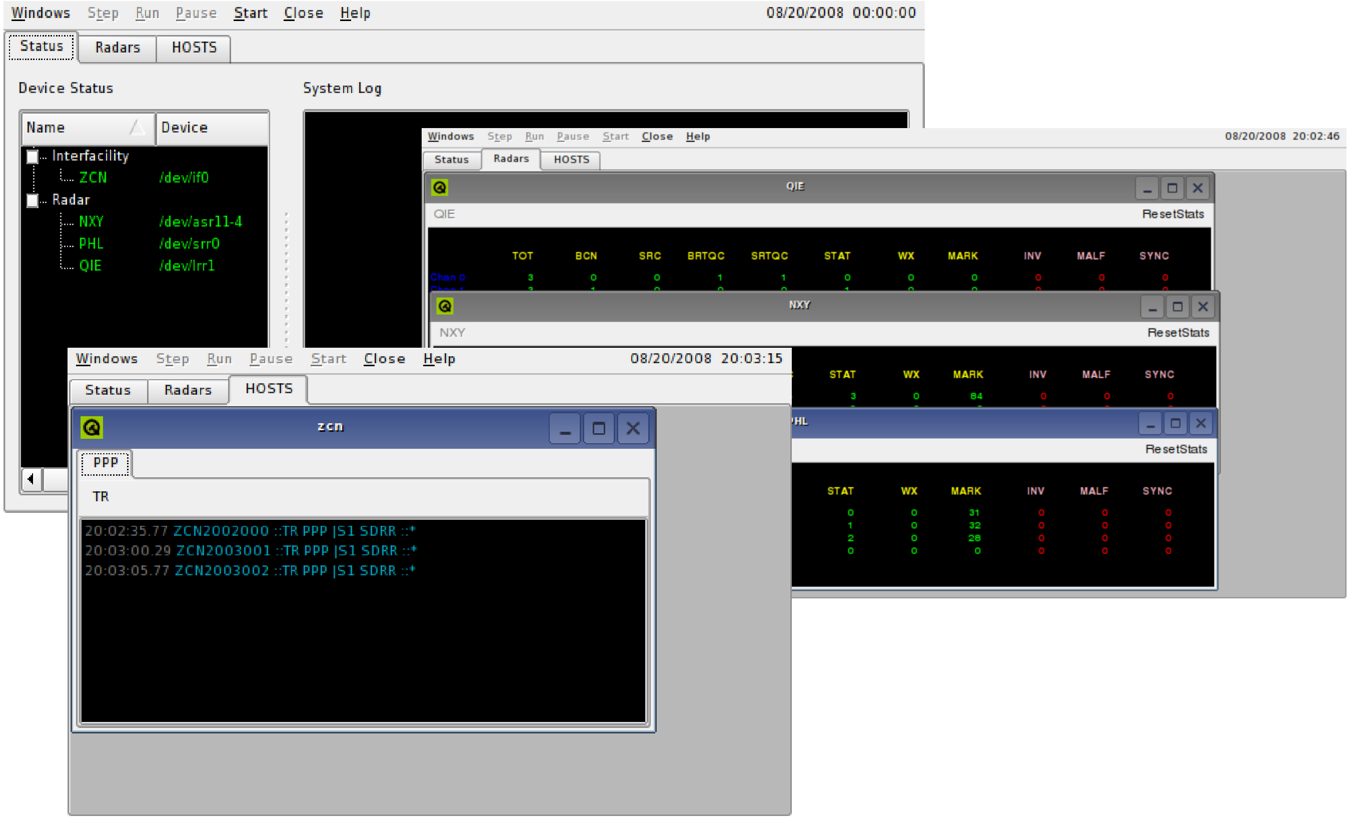
File name:
File type: All Files (*)

Look in: /home/stars/SIRS-PHL-scenario/

..
srv
playback

File name: playback
File type: All Files (*)

After the script is selected and the **Open** button is clicked, the SDRR xtest is launched. The **SDRR** interface loads the defined **devices** and **facilities** from the selected script. It looks just like a SDRR playback interface (screenshots on the next page). The way it works is that **XTEST** utility is running the **firs2sdrr** utility on the fly while playing the scenario. This is fine, but better performance can be achieved by using the **firs2sdrr** utility first.



08/20/2008 00:00:00

Name	Device
Interfacility	
ZCN	/dev/if0
Radar	
NXY	/dev/asr11-4
PHL	/dev/srr0
QIE	/dev/lrr1

08/20/2008 20:02:46

QIE

	TOT	BCN	SRC	BRTQC	SRTQC	STAT	WX	MARK	INV	MALF	SYNC
Chan 0	3	0	0	1	1	0	0	0	0	0	0

NXY

STAT	WX	MARK	INV	MALF	SYNC
3	0	84	0	0	0

PHL

STAT	WX	MARK	INV	MALF	SYNC
0	0	31	0	0	0
1	0	32	0	0	0
2	0	28	0	0	0
0	0	0	0	0	0

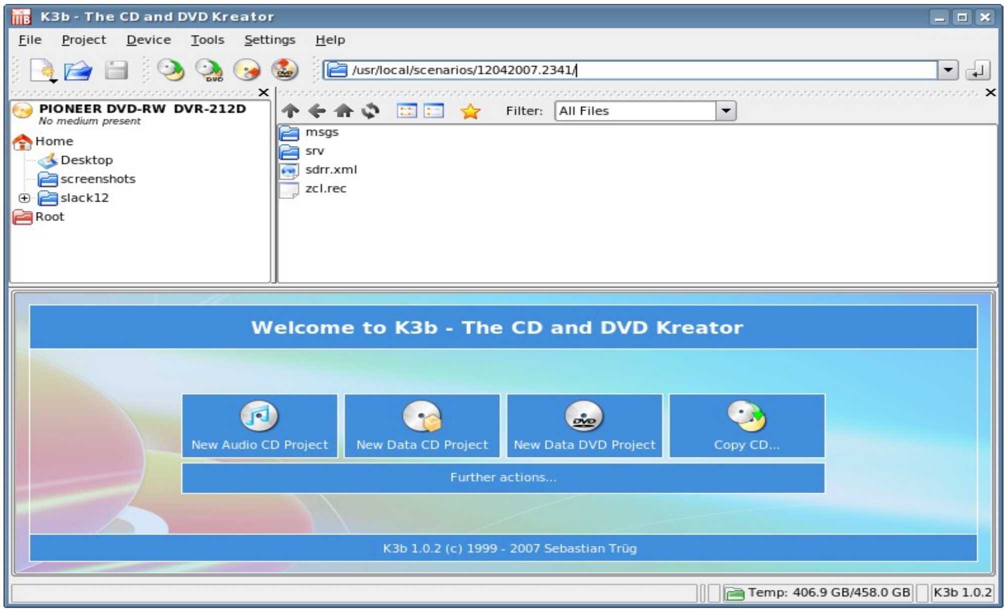
08/20/2008 20:03:15

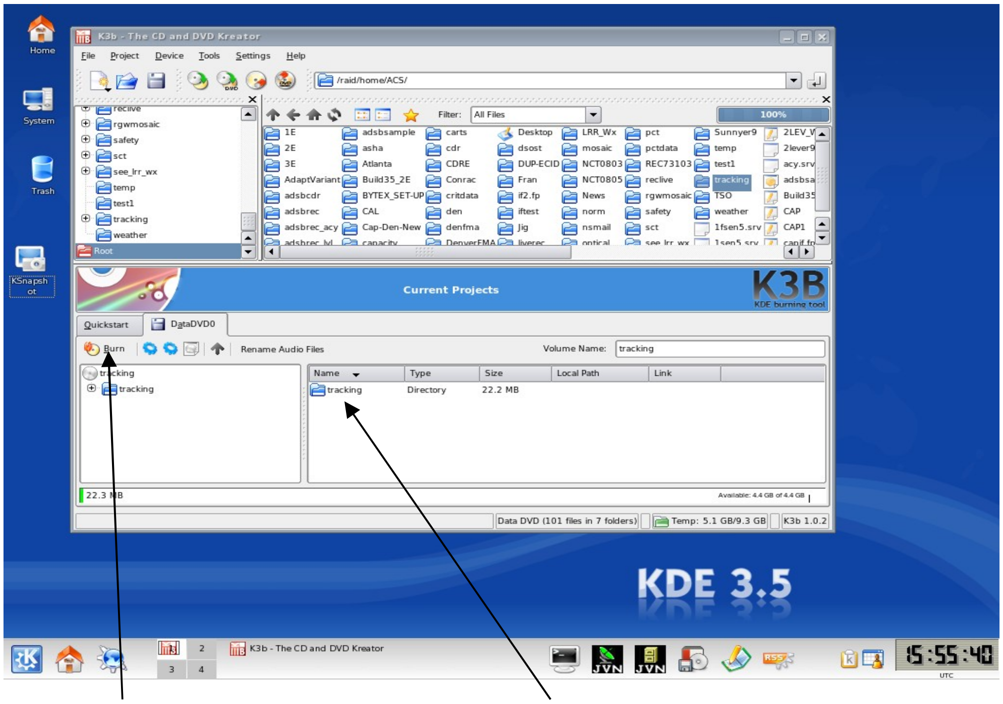
```
20:02:35.77 ZCN2002000 ::TR PPP |S1 SDRR ::*
20:03:00.29 ZCN2003001 ::TR PPP |S1 SDRR ::*
20:03:05.77 ZCN2003002 ::TR PPP |S1 SDRR ::*
```

7.21. KDE’s K3b CD & DVD Creator Utility

This utility is provided by the KDE desktop environment and allows the user to drag and drop files which need to be copied onto CD or DVD.

It is launched from the desktop.

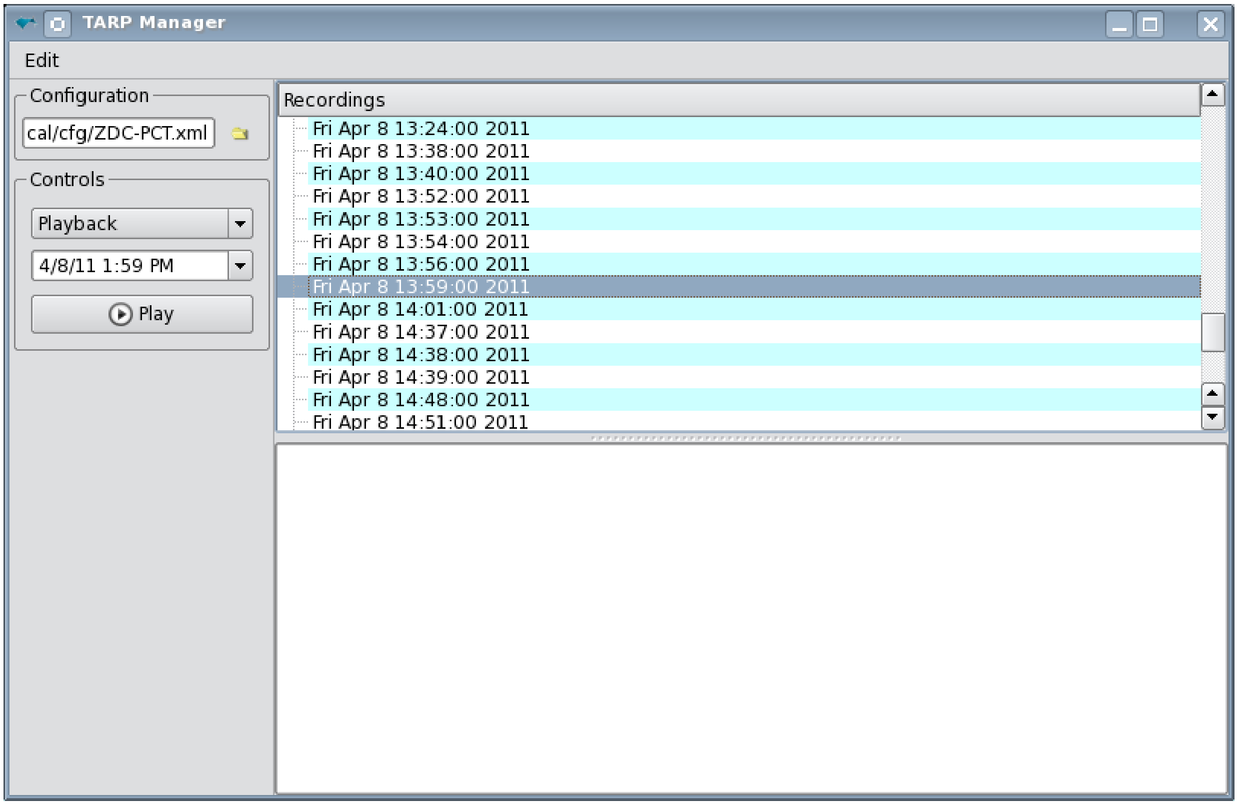




After selecting the project type, use the browser to locate the desired files and folders. Drag them into the window at the bottom. When they are ready for recording the “Burn” button on the left will no longer be grayed out and should appear in full color. Insert blank media and click the “Burn” button. K3b program will tell you when it is finished creating new disc.

7.22. TARP

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



Configuration: cal/cfg/ZDC-PCT.xml

Controls: Playback

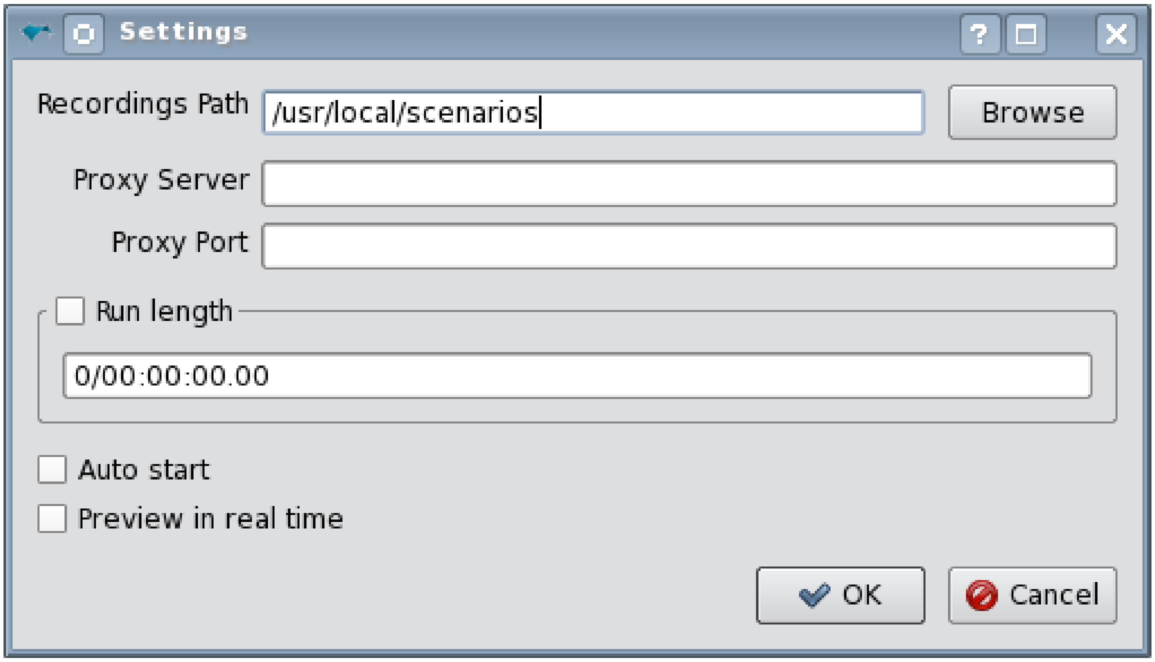
4/8/11 1:59 PM

Recordings:

- Fri Apr 8 13:24:00 2011
- Fri Apr 8 13:38:00 2011
- Fri Apr 8 13:40:00 2011
- Fri Apr 8 13:52:00 2011
- Fri Apr 8 13:53:00 2011
- Fri Apr 8 13:54:00 2011
- Fri Apr 8 13:56:00 2011
- Fri Apr 8 13:59:00 2011
- Fri Apr 8 14:01:00 2011
- Fri Apr 8 14:37:00 2011
- Fri Apr 8 14:38:00 2011
- Fri Apr 8 14:39:00 2011
- Fri Apr 8 14:48:00 2011
- Fri Apr 8 14:51:00 2011

Select the desired configuration to be used 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 user can select *Playback/Preview* from the drop down box under *Controls* section. When preview is selected the user will have the ability to specify an option 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.

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



Recordings Path /usr/local/scenarios

Proxy Server

Proxy Port

□ Run length

0/00:00:00.00

□ Auto start

□ Preview in real time

Recordings Path: This is the location of the recordings.
Proxy Server: Specify the proxy server if available.
Proxy Port: Specify the proxy port if available.
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.

Appendix A. Environment Variables

SDRR uses a number of environment variables to specify various input and output locations. They are listed below with typical values shown in parenthesis.

SDRR_SCENARIO_PATH: Location of SDRR scenarios (/usr/local/scenarios)

SDRR: Location of SDRR build (/usr/local/sdrr.2.2.6)

SDRR_USER_PATH: Location of user home directories (/home)

SDRR_CONFIG_PATH: Location of SDRR configuration files (/usr/local/cfg)

SDRR_LOG_PATH: Location where SDRR log files will be written (/usr/local/log)

Appendix B. Scenario File Format

The Scenario file (called sdrr.xml) defines the files to be read by SDRR and defines parameters for the scenario.

Sim - Defines options for running SIRDS.

Required Parameters:

name: Name of the scenario.

Optional Parameters:

starttime: Specifies scenario start time.

giStart: Start upon receiving GI message from HOST (default is 0).

autotu: Enables the automatic TU generation for this facility. Defines the number of seconds to wait before sending an automatic TU after sending a TI message (default is 0).

autoTa: Enables the automatic TA response for this facility. Defines the number of seconds to wait before sending an automatic TA response to TI message (default is 0).

rsi: Specifies a list of RSI-tagged messages to be injected.

logfile: Specifies the location of the SIRDS log file.

utc: Enables synchronizing the scenario to system time (default is 0).

comments

Free-form scenario comments.

Radar - Defines radar file inputs.

Required Parameters:

name: Name of the radar.

insrv: Specifies the location of the binary radar file.

Msgs - Defines message file inputs.

Required Parameters:

file: Specifies the location of the msgs file.

Optional Parameters:

facility: Specifies the facility used in the msgs file.

Tracks - Defines a SDRR track file. The track file is normally generated by GSGT and is used by SIRDS in generating TU messages. If no file is specified, SDRR will extrapolate the position information from the TI message.

Required Parameters:

file: Specifies the location of the tracks file.

Tgtctl - Defines customized responses to specified messages.

Required Parameters:

file: Specifies the location of the tgtctl file.

Appendix C. Sample Scenario File

```
<sim name="example" starttime="00:00:00.00" giStart="0" autotu="1" autoTa="10" utc="0" >
   <comments> SDRR Scenario File Example </comments>
   <msgs file="msgs/msgs.xml" />
   <radar name="acy" srv="srv/acy.srv" wx="srv/acy.wx" />
   <radar name="qie" srv="srv/qie.srv" />
   <radar name="qpl" srv="srv/qpl.srv" />
   <radar name="phl" srv="srv/phl.srv" wx="srv/phl.wx" />
   <radar name="nxy" srv="srv/nxy.srv" wx="srv/nxy.wx" />
   <radar name="qvh" srv="srv/qvh.srv" />
   <tgtctl file="repctl/tgtctl.xml" />
</sim>
```

Appendix D. Configuration File Format

The configuration file is an xml file which defines the facilities and/or sensors to be simulated.

Host

Defines a NAS Host facility. This facility has 1 interfacility device per io thread.

Attributes:

name: Used as the title of the display window for this facility, and is matched with the "src" field in the scenario file format for messages (see Scenario File Format)

facName: Defines the 3-letter identifier to be used on the wire to identify this site. If not defined, it defaults to the first 3 characters of "name".

facID: Defines the 1-letter identifier to be used on the wire to identify this site. If not defined, defaults to the 3rd character of "facName".

autoTA: Enables the automatic TA response for this facility. Defines the number of seconds to wait before sending an automatic TA response to TI message (default is 0).

autoTR: Enables periodic transmission of TR message, defines the interval in seconds (default is 0).

Threads:

hostio: defines an interface to a NAS host.

Attributes:

name: Used as the title of the display window for this facility, and is matched with the "dest" field in the scenario file format for messages (see Scenario File Format)

facName: Defines the 3-letter identifier to be used on the wire to identify this site. If not defined, it defaults to the first 3 characters of "name".

facID: Defines the 1-letter identifier to be used on the wire to identify this site. If not defined, defaults to the 3rd character of "facName".

device: Device or file name.

txclock: The baud rate of the transmit side of the assigned device. 0 means accept the transmit clock from the DCE (default is 2400).

rxclock: The baud rate of the receive side of the assigned device. 0 means accept the receive clock from the DCE (default is 0).

tangent: Lat/Long Point Of Tangency

org: X/Y value used for dynamic interfacility messaging

magdev: magnetic deviation, in degrees

artsio: defines an interface to a Terminal facility.

Attributes:

name: Used as the title of the display window for this facility, and is matched with the "dest" field in the scenario file format for messages (see Scenario File Format)

facName: Defines the 3-letter identifier to be used on the wire to identify this site. If not defined, it defaults to the first 3 characters of “name”.

facID: Defines the 1-letter identifier to be used on the wire to identify this site. If not defined, defaults to the 3rd character of “facName”.

device: Device or file name.

txclock: The baud rate of the transmit side of the assigned device. 0 means accept the transmit clock from the DCE (default is 2400).

rxclock: The baud rate of the receive side of the assigned device. 0 means accept the receive clock from the DCE (default is 0).

tangent: Lat/Long Point Of Tangency

org: X/Y value used for dynamic interfacility messaging

magdev: magnetic deviation, in degrees

Arts

Defines a terminal (ARTS/STARS) facility. This facility has 1 interfacility device associated with it, with 1 or more io threads to handle communications to other facilities which are all routed through the single interfacility device.

Attributes:

device: Device or file name.

name: Used as the title of the display window for this facility, and is matched with the “src” field in the scenario file format for messages (see Scenario File Format)

facName: Defines the 3-letter identifier to be used on the wire to identify this site. If not defined, it defaults to the first 3 characters of “name”.

facID: Defines the 1-letter identifier to be used on the wire to identify this site. If not defined, defaults to the 3rd character of “facName”.

autoTA: Enables the automatic TA response for this facility. Defines the number of seconds to wait before sending an automatic TA response to TI message (default is 0).

autoTR: Enables periodic transmission of TR message, defines the interval in seconds (default is 0).

Threads:

hostio: defines an interface to a NAS host.

Attributes:

name: Used as the title of the display window for this facility, and is matched with the “dest” field in the scenario file format for messages (see Scenario File Format)

facName: Defines the 3-letter identifier to be used on the wire to identify this site. If not defined, it defaults to the first 3 characters of “name”.

facID: Defines the 1-letter identifier to be used on the wire to identify this site. If not defined, defaults to the 3rd character of “facName”.

txclock: The baud rate of the transmit side of the assigned device. 0 means accept the transmit clock from the DCE (default is 2400).

rxclock: The baud rate of the receive side of the assigned device. 0 means accept the receive clock from the DCE (default is 0).

tangent: Lat/Long Point Of Tangency

org: X/Y value used for dynamic interfacility messaging

magdev: magnetic deviation, in degrees

artsio: defines an interface to a Terminal facility.

Attributes:

name: Used as the title of the display window for this facility, and is matched with the "dest" field in the scenario file format for messages (see Scenario File Format)

facName: Defines the 3-letter identifier to be used on the wire to identify this site. If not defined, it defaults to the first 3 characters of "name".

facID: Defines the 1-letter identifier to be used on the wire to identify this site. If not defined, defaults to the 3rd character of "facName".

txclock: The baud rate of the transmit side of the assigned device. 0 means accept the transmit clock from the DCE (default is 2400).

rxclock: The baud rate of the receive side of the assigned device. 0 means accept the receive clock from the DCE (default is 0).

tangent: Lat/Long Point Of Tangency

org: X/Y value used for dynamic interfacility messaging

magdev: magnetic deviation, in degrees

Radar

Defines a radar sensor.

Attributes:

name: Used as the title of the display window for this facility, and is matched with the "src" field in the scenario file format for messages (see Scenario File Format)

type: radar type (defaults to LRR)

device: Device or file name.

chans: number of channels (default is 0, which means auto-determine based on type)

magdev: magnetic deviation of the radar, in degrees

parrots: defines the beacon code(s) squawked by parrot targets

scantime: amount of time the radar takes to complete one sweep, in seconds

elev: elevation of the radar, in feet above MSL

pos: lat/long position of the radar

Svol

Defines a service volume for ADS-B.

Attributes:

name: Used as the title of the display window for this facility, and is matched with the "src" field in the scenario file format for messages (see Scenario File Format)

pos: lat/long position

Threads:

stream: Defines the type(s) of message supported.

Attributes:

name: Used as the title of the display window, and is matched with the “src” field in the scenario file format for messages (see Scenario File Format). It must be one of: UAT, 1090, Equip, SVol, or SDP.

device: Device or file name.

Mlat

Defines ########

Attributes:

name: Used as the title of the display window for this facility, and is matched with the “src” field in the scenario file format for messages (see Scenario File Format)

ttl: defines #######

Etms

Defines an ETMS facility.

Attributes:

starsid: defines a three-letter ID for use by a STARS facility

name: Used as the title of the display window for this facility, and is matched with the “src” field in the scenario file format for messages (see Scenario File Format)

device: Device or file name.

rate: baud rate

Apex

Defines an APEX device. Each I/O thread is assigned its own device.

Attributes:

autoHello: flag to periodically send Hello message. (defaults to 1 – send)

Threads:

rdpio: creates a RDP interface.

Attributes:

name: Used as the title of the display window for this facility, and is matched with the “dest” field in the scenario file format for messages (see Scenario File Format)

facName: Defines the 3-letter identifier to be used on the wire to identify this site. If not defined, it defaults to the first 3 characters of “name”.

facID: Defines the 1-letter identifier to be used on the wire to identify this site. If not defined, defaults to the 3rd character of “facName”.

device: Device or file name.

txclock: The baud rate of the transmit side of the assigned device. 0 means accept the transmit clock from the DCE (default is 2400).

rxclock: The baud rate of the receive side of the assigned device. 0 means accept the receive clock from the DCE (default is 0).

debug: Enables display of JCAB protocol messages (default is 0).

penqTime: Specifies the time interval between p-ENQ messages in the JCAB protocol (default is 1.0).
priority: flag to indicate high priority when negotiating for JCAB transmissions. (defaults to 0 (low priority))

prefix: JCAB protocol prefix for this interface (defaults to “0”).

remotePrefix: JCAB protocol prefix of the remote end of this interface (defaults to “0”).

fdpio: creates a FDP interface.

Attributes:

name: Used as the title of the display window for this facility, and is matched with the “dest” field in the scenario file format for messages (see Scenario File Format)

facName: Defines the 3-letter identifier to be used on the wire to identify this site. If not defined, it defaults to the first 3 characters of “name”.

facID: Defines the 1-letter identifier to be used on the wire to identify this site. If not defined, defaults to the 3rd character of “facName”.

device: Device or file name.

txclock: The baud rate of the transmit side of the assigned device. 0 means accept the transmit clock from the DCE (default is 2400).

rxclock: The baud rate of the receive side of the assigned device. 0 means accept the receive clock from the DCE (default is 0).

debug: Enables display of JCAB protocol messages (default is 0).

penqTime: Specifies the time interval between p-ENQ messages in the JCAB protocol (default is 1.0).
priority: flag to indicate high priority when negotiating for JCAB transmissions. (defaults to 0 (low priority))

prefix: JCAB protocol prefix for this interface (defaults to “0”).

remotePrefix: JCAB protocol prefix of the remote end of this interface (defaults to “0”).

artsfio: creates an interface to an ArtsF facility.

Attributes:

name: Used as the title of the display window for this facility, and is matched with the “dest” field in the scenario file format for messages (see Scenario File Format)

facName: Defines the 3-letter identifier to be used on the wire to identify this site. If not defined, it defaults to the first 3 characters of “name”.

facID: Defines the 1-letter identifier to be used on the wire to identify this site. If not defined, defaults to the 3rd character of “facName”.

device: Device or file name.

txclock: The baud rate of the transmit side of the assigned device. 0 means accept the transmit clock from the DCE (default is 2400).

rxclock: The baud rate of the receive side of the assigned device. 0 means accept the receive clock from the DCE (default is 0).

debug: Enables display of JCAB protocol messages (default is 0).

penqTime: Specifies the time interval between p-ENQ messages in the JCAB protocol (default is 1.0).
priority: flag to indicate high priority when negotiating for JCAB transmissions. (defaults to 0 (low priority))

prefix: JCAB protocol prefix for this interface (defaults to “0”).

remotePrefix: JCAB protocol prefix of the remote end of this interface (defaults to “0”).

starsio: creates an interface to a STARS facility.

Attributes:

name: Used as the title of the display window for this facility, and is matched with the “dest” field in the scenario file format for messages (see Scenario File Format)

facName: Defines the 3-letter identifier to be used on the wire to identify this site. If not defined, it defaults to the first 3 characters of “name”.

facID: Defines the 1-letter identifier to be used on the wire to identify this site. If not defined, defaults to the 3rd character of “facName”.

device: Device or file name.

txclock: The baud rate of the transmit side of the assigned device. 0 means accept the transmit clock from the DCE (default is 2400).

rxclock: The baud rate of the receive side of the assigned device. 0 means accept the receive clock from the DCE (default is 0).

tangent: Lat/Long Point Of Tangency used to convert X/Y in STARS TI mgs to/from Lat/Long in JNAS HI msgs.

ArtsF

Defines a Japanese terminal facility. This facility has 1 interfacility device per io thread.

Attributes:

autoHello: flag to periodically send Hello message. (defaults to 1 – send)

name: Used as the title of the display window for this facility, and is matched with the “src” field in the scenario file format for messages (see Scenario File Format)

facName: Defines the 3-letter identifier to be used on the wire to identify this site. If not defined, it defaults to the first 3 characters of “name”.

Threads:

rdpio: creates a RDP interface.

Attributes:

name: Used as the title of the display window for this facility, and is matched with the “dest” field in the scenario file format for messages (see Scenario File Format)

facName: Defines the 3-letter identifier to be used on the wire to identify this site. If not defined, it defaults to the first 3 characters of “name”.

facID: Defines the 1-letter identifier to be used on the wire to identify this site. If not defined, defaults to the 3rd character of “facName”.

device: Device or file name.

txclock: The baud rate of the transmit side of the assigned device. 0 means accept the transmit clock from the DCE (default is 2400).

rxclock: The baud rate of the receive side of the assigned device. 0 means accept the receive clock from the DCE (default is 0).

debug: Enables display of JCAB protocol messages (default is 0).

penqTime: Specifies the time interval between p-ENQ messages in the JCAB protocol (default is 1.0).
priority: flag to indicate high priority when negotiating for JCAB transmissions. (defaults to 1 (high priority))

prefix: JCAB protocol prefix for this interface (defaults to “0”).

remotePrefix: JCAB protocol prefix of the remote end of this interface (defaults to “0”).

fdpio: creates a FDP interface.

Attributes:

name: Used as the title of the display window for this facility, and is matched with the “dest” field in the scenario file format for messages (see Scenario File Format)

facName: Defines the 3-letter identifier to be used on the wire to identify this site. If not defined, it defaults to the first 3 characters of “name”.

facID: Defines the 1-letter identifier to be used on the wire to identify this site. If not defined, defaults to the 3rd character of “facName”.

device: Device or file name.

txclock: The baud rate of the transmit side of the assigned device. 0 means accept the transmit clock from the DCE (default is 2400).

rxclock: The baud rate of the receive side of the assigned device. 0 means accept the receive clock from the DCE (default is 0).

debug: Enables display of JCAB protocol messages (default is 0).

penqTime: Specifies the time interval between p-ENQ messages in the JCAB protocol (default is 1.0).
priority: flag to indicate high priority when negotiating for JCAB transmissions. (defaults to 1 (high priority))

prefix: JCAB protocol prefix for this interface (defaults to “0”).

remotePrefix: JCAB protocol prefix of the remote end of this interface (defaults to “0”).

artsfio: creates an interface to an ArtsF facility.

Attributes:

name: Used as the title of the display window for this facility, and is matched with the “dest” field in the scenario file format for messages (see Scenario File Format)

facName: Defines the 3-letter identifier to be used on the wire to identify this site. If not defined, it defaults to the first 3 characters of “name”.

facID: Defines the 1-letter identifier to be used on the wire to identify this site. If not defined, defaults to the 3rd character of “facName”.

device: Device or file name.

txclock: The baud rate of the transmit side of the assigned device. 0 means accept the transmit clock from the DCE (default is 2400).

rxclock: The baud rate of the receive side of the assigned device. 0 means accept the receive clock from the DCE (default is 0).

debug: Enables display of JCAB protocol messages (default is 0).

penqTime: Specifies the time interval between p-ENQ messages in the JCAB protocol (default is 1.0).
priority: flag to indicate high priority when negotiating for JCAB transmissions. (defaults to 1 (high priority))

prefix: JCAB protocol prefix for this interface (defaults to "0").

remotePrefix: JCAB protocol prefix of the remote end of this interface (defaults to "0").

Jacc

Defines a Japanese Host facility. Each I/O thread is assigned its own device.

Attributes:

autoHello: flag to periodically send Hello message. (defaults to 1 – send)

Threads:

rdpio: creates an RDP interface.

Attributes:

name: Used as the title of the display window for this facility, and is matched with the "dest" field in the scenario file format for messages (see Scenario File Format)

facName: Defines the 3-letter identifier to be used on the wire to identify this site. If not defined, it defaults to the first 3 characters of "name".

facID: Defines the 1-letter identifier to be used on the wire to identify this site. If not defined, defaults to the 3rd character of "facName".

device: Device or file name.

txclock: The baud rate of the transmit side of the assigned device. 0 means accept the transmit clock from the DCE (default is 2400).

rxclock: The baud rate of the receive side of the assigned device. 0 means accept the receive clock from the DCE (default is 0).

debug: Enables display of JCAB protocol messages (default is 0).

penqTime: Specifies the time interval between p-ENQ messages in the JCAB protocol (default is 1.0).
priority: flag to indicate high priority when negotiating for JCAB transmissions. (defaults to 0 (low priority))

prefix: JCAB protocol prefix for this interface (defaults to "0").

remotePrefix: JCAB protocol prefix of the remote end of this interface (defaults to "0").

fdpio: creates an FDP interface.

Attributes:

name: Used as the title of the display window for this facility, and is matched with the "dest" field in the scenario file format for messages (see Scenario File Format)

facName: Defines the 3-letter identifier to be used on the wire to identify this site. If not defined, it defaults to the first 3 characters of "name".

facID: Defines the 1-letter identifier to be used on the wire to identify this site. If not defined, defaults to the 3rd character of “facName”.

device: Device or file name.

txclock: The baud rate of the transmit side of the assigned device. 0 means accept the transmit clock from the DCE (default is 2400).

rxclock: The baud rate of the receive side of the assigned device. 0 means accept the receive clock from the DCE (default is 0).

debug: Enables display of JCAB protocol messages (default is 0).

penqTime: Specifies the time interval between p-ENQ messages in the JCAB protocol (default is 1.0).
priority: flag to indicate high priority when negotiating for JCAB transmissions. (defaults to 1 (high priority))

prefix: JCAB protocol prefix for this interface (defaults to “0”).

remotePrefix: JCAB protocol prefix of the remote end of this interface (defaults to “0”).

artsfio: Creates an interface to an ArtsF facility.

Attributes:

name: Used as the title of the display window for this facility, and is matched with the “dest” field in the scenario file format for messages (see Scenario File Format)

facName: Defines the 3-letter identifier to be used on the wire to identify this site. If not defined, it defaults to the first 3 characters of “name”.

facID: Defines the 1-letter identifier to be used on the wire to identify this site. If not defined, defaults to the 3rd character of “facName”.

device: Device or file name.

txclock: The baud rate of the transmit side of the assigned device. 0 means accept the transmit clock from the DCE (default is 2400).

rxclock: The baud rate of the receive side of the assigned device. 0 means accept the receive clock from the DCE (default is 0).

debug: Enables display of JCAB protocol messages (default is 0).

penqTime: Specifies the time interval between p-ENQ messages in the JCAB protocol (default is 1.0).
priority: flag to indicate high priority when negotiating for JCAB transmissions. (defaults to 1 (high priority))

prefix: JCAB protocol prefix for this interface (defaults to “0”).

remotePrefix: JCAB protocol prefix of the remote end of this interface (defaults to “0”).

Pidp

Defines a DoD terminal (PIDP) facility. This facility has 1 interfacility device associated with it, with 1 or more io threads to handle communications to other facilities which are all routed through the single interfacility device.

Threads:

hostio: defines an interface to a NAS host.

artsio: defines an interface to a Terminal facility.

starsio: creates an interface to a STARS facility.

Appendix E. Sample Configuration File

Standard configuration example:

```
<sources>
  <comments>Sample Config File</comments>
  <host name="zdc" facName="zcw" facID="w" autoTR="0">
    <hostio name="zbw" device="pipe:zbw-zdc" facName="zcb" facID="b" tangent="+43:48:33.98,-
072:26:24.00" org="-264.50,-344.25"/>
  </host>
  <host name="zny" facName="zcn" facID="n" autoTR="0">
    <hostio name="zbw" device="pipe:zbw-zny" facName="zcb" facID="b" tangent="+43:48:33.98,-
072:26:24.00" org="-264.50,-344.25"/>
  </host>
  <host name="zob" facName="zcc" facID="c" autoTR="0">
    <hostio name="zbw" device="pipe:zbw-zob" facName="zcb" facID="b" tangent="+43:48:33.98,-
072:26:24.00" org="-264.50,-344.25"/>
  </host>
  <arts name="aca" device="pipe:zbw-aca" facName="aca" autoTR="0">
    <hostio name="zbw" facName="zcb" magdev="-15.80" tangent="+41:14:55.00,-070:03:39.00"/>
    <artsio name="fma" magdev="-15.60" tangent="+41:39:44.00,-070:31:21.00"/>
  </arts>
  <arts name="ala" device="pipe:zbw-ala" facName="ala" autoTR="0">
    <hostio name="zbw" facName="zcb" magdev="-14.00" tangent="+42:44:30.00,-073:49:13.00"/>
    <artsio name="bda" magdev="-15.00" tangent="+41:56:19.00,-072:40:57.00"/>
    <artsio name="rma" magdev="-13.00" tangent="+43:13:41.00,-075:25:26.00"/>
  </arts>
  <arts name="bda" device="pipe:zbw-bda" facName="bda" autoTR="0">
    <hostio name="zbw" facName="zcb" magdev="-15.00" tangent="+41:56:19.00,-072:40:57.00"/>
    <artsio name="ala" magdev="-14.00" tangent="+42:44:30.00,-073:49:13.00"/>
    <artsio name="boa" magdev="-16.00" tangent="+42:20:55.00,-071:00:22.00"/>
    <artsio name="nca" magdev="-15.00" tangent="+41:43:14.00,-071:35:53.00"/>
  </arts>
  <arts name="bga" device="pipe:zbw-bga" facName="bga" autoTR="0">
    <hostio name="zbw" facName="zcb" magdev="-17.70" tangent="+44:48:21.00,-068:50:20.00"/>
    <artsio name="pwa" magdev="-17.00" tangent="+43:49:42.00,-070:15:58.00"/>
  </arts>
  <arts name="boa" device="pipe:zbw-boa" facName="boa" autoTR="0">
    <hostio name="zbw" facName="zcb" magdev="-16.00" tangent="+42:20:55.00,-071:00:22.00"/>
    <artsio name="fma" magdev="-15.60" tangent="+41:39:44.00,-070:31:21.00"/>
    <artsio name="bda" magdev="-15.00" tangent="+41:56:19.00,-072:40:57.00"/>
    <artsio name="nca" magdev="-15.00" tangent="+41:43:14.00,-071:35:53.00"/>
    <artsio name="pwa" magdev="-17.00" tangent="+43:49:42.00,-070:15:58.00"/>
  </arts>
  <arts name="bta" device="pipe:zbw-bta" facName="bta" autoTR="0">
    <hostio name="zbw" facName="zcb" magdev="-16.00" tangent="+44:28:02.00,-073:09:02.00"/>
  </arts>
  <arts name="fma" device="pipe:zbw-fma" facName="fma" autoTR="0">
    <hostio name="zbw" facName="zcb" magdev="-15.60" tangent="+41:39:44.00,-070:31:21.00"/>
    <artsio name="aca" magdev="-15.80" tangent="+41:14:55.00,-070:03:39.00"/>
    <artsio name="nca" magdev="-15.00" tangent="+41:43:14.00,-071:35:53.00"/>
    <artsio name="boa" magdev="-16.00" tangent="+42:20:55.00,-071:00:22.00"/>
  </arts>
  <arts name="gta" device="pipe:zbw-gta" facName="gta" autoTR="0">
    <hostio name="zbw" facName="zcb" magdev="-14.00" tangent="+44:02:43.00,-075:42:57.00"/>
    <artsio name="rma" magdev="-13.00" tangent="+43:13:41.00,-075:25:26.00"/>
    <artsio name="sya" magdev="-13.00" tangent="+43:06:44.00,-076:06:20.00"/>
  </arts>
  <arts name="nca" device="pipe:zbw-nca" facName="nca" autoTR="0">
    <hostio name="zbw" facName="zcb" magdev="-15.00" tangent="+41:43:14.00,-071:35:53.00"/>
    <artsio name="boa" magdev="-16.00" tangent="+42:20:55.00,-071:00:22.00"/>
    <artsio name="fma" magdev="-15.60" tangent="+41:39:44.00,-070:31:21.00"/>
    <artsio name="bda" magdev="-15.00" tangent="+41:56:19.00,-072:40:57.00"/>
  </arts>
  <arts name="pwa" device="pipe:zbw-pwa" facName="pwa" autoTR="0">
    <hostio name="zbw" facName="zcb" magdev="-17.00" tangent="+43:49:42.00,-070:15:58.00"/>
    <artsio name="bga" magdev="-17.70" tangent="+44:48:21.00,-068:50:20.00"/>
    <artsio name="boa" magdev="-16.00" tangent="+42:20:55.00,-071:00:22.00"/>
  </arts>
  <arts name="rma" device="pipe:zbw-rma" facName="rma" autoTR="0">
    <hostio name="zbw" facName="zcb" magdev="-13.00" tangent="+43:13:41.00,-075:25:26.00"/>
    <artsio name="gta" magdev="-14.00" tangent="+44:02:43.00,-075:42:57.00"/>
    <artsio name="ala" magdev="-14.00" tangent="+42:44:30.00,-073:49:13.00"/>
    <artsio name="sya" magdev="-13.00" tangent="+43:06:44.00,-076:06:20.00"/>
  </arts>
  <arts name="sya" device="pipe:zbw-sya" facName="sya" autoTR="0">
    <hostio name="zbw" facName="zcb" magdev="-13.00" tangent="+43:06:44.00,-076:06:20.00"/>
    <artsio name="gta" magdev="-14.00" tangent="+44:02:43.00,-075:42:57.00"/>
    <artsio name="rma" magdev="-13.00" tangent="+43:13:41.00,-075:25:26.00"/>
  </arts>
  <radar name="bos" device="/dev/srr0" type="asr9-modes" magdev="-16.00" parrots="1274,1273" scantime="4.75" elev="0.00" pos="+42:20:54.80,-071:00:22.00" color="#446a40"/>
  <radar name="qea" device="/dev/lrr1" type="arsr2" parrots="1271" scantime="12.00" elev="0.00" pos="+42:02:02.40,-070:03:11.10" color="#324c0d"/>
  <radar name="qvh" device="/dev/lrr2" type="arsr4" parrots="1257" scantime="12.00" elev="0.00" pos="+40:52:43.00,-072:41:14.00" color="#b01488"/>
  <radar name="pvd" device="/dev/asr11-3" asr11_dup="1" type="asr11" parrots="1272" scantime="4.80" elev="0.00" pos="+41:43:14.00,-071:35:53.00" color="#495e29"/>
  <radar name="mht" device="hostname:/dev/srr0" type="asr9-ibi" magdev="-16.00" parrots="1270,1275" scantime="4.75" elev="0.00" pos="+42:57:00.10,-071:17:57.20" color="#012172"/>
  <radar name="bdl" device="(multi:eth1:224.100.250.8/8250+multi:eth2:224.100.250.9/9250)" type="asr11" scantime="4.80" format="ast" ttl="19" elev="0.00" pos="+41:56:18.80,-072:40:57.00" color="#065f1c"/>
  <mlat name="mlt" device="(multi:eth1:224.100.150.8/8150+multi:eth2:224.100.150.9/9150)" ttl="19" color="#7f725f"/>
   <svol name="ads" color="#067125" pos="+42:21:00.00,-071:01:00.00">
    <stream name="UAT" device="multi:eth0:239.160.28.30/59900" />
    <stream name="1090" device="multi:eth0:239.160.28.30/59901" />
    <stream name="Equip" device="multi:eth0:239.160.28.30/59902" />
    <stream name="SVol" device="multi:eth0:239.160.28.30/59903" />
    <stream name="SDP" device="multi:eth0:239.160.28.30/59904" />
  </svol>
  <host name="zbw" facName="zcb" facID="b" autoTR="0" autoTA="10" tangent="+43:48:33.98,-072:26:24.00" org="-264.500,-344.250">
    <artsio name="aca" device="pipe:zbw-aca" facName="aca" magdev="-15.80" tangent="+41:14:55.00,-070:03:39.00"/>
    <artsio name="ala" device="pipe:zbw-ala" facName="ala" magdev="-14.00" tangent="+42:44:30.00,-073:49:13.00"/>
    <artsio name="bda" device="pipe:zbw-bda" facName="bda" magdev="-15.00" tangent="+41:56:19.00,-072:40:57.00"/>
    <artsio name="bga" device="pipe:zbw-bga" facName="bga" magdev="-17.70" tangent="+44:48:21.00,-068:50:20.00"/>
    <artsio name="boa" device="/dev/if0" facName="boa" magdev="-16.00" tangent="+42:20:55.00,-071:00:22.00" txclock="2400" rxclock="2400" />
    <artsio name="bta" device="pipe:zbw-bta" facName="bta" magdev="-16.00" tangent="+44:28:02.00,-073:09:02.00"/>
    <artsio name="fma" device="pipe:zbw-fma" facName="fma" magdev="-15.60" tangent="+41:39:44.00,-070:31:21.00"/>
    <artsio name="gta" device="pipe:zbw-gta" facName="gta" magdev="-14.00" tangent="+44:02:43.00,-075:42:57.00"/>
    <artsio name="nca" device="pipe:zbw-nca" facName="nca" magdev="-15.00" tangent="+41:43:14.00,-071:35:53.00"/>
    <artsio name="pwa" device="pipe:zbw-pwa" facName="pwa" magdev="-17.00" tangent="+43:49:42.00,-070:15:58.00"/>
    <artsio name="rma" device="pipe:zbw-rma" facName="rma" magdev="-13.00" tangent="+43:13:41.00,-075:25:26.00"/>
    <artsio name="sya" device="pipe:zbw-sya" facName="sya" magdev="-13.00" tangent="+43:06:44.00,-076:06:20.00"/>
    <hostio name="zdc" device="pipe:zbw-zdc" facName="zcw" org="-334.375,-400.500" tangent="+36:51:06.00,-075:36:08.00"/>
    <hostio name="zny" device="pipe:zbw-zny" facName="zcn" org="-525.750,-642.875" tangent="+37:40:56.00,-070:43:11.00">
      <externalArts localName="eee" remoteName="eee"/>
      <externalArts localName="www" remoteName="www"/>
      <externalArts localName="nnn" remoteName="nnn"/>
      <externalArts localName="bbb" remoteName="bbb"/>
    </hostio>
    <hostio name="zob" device="pipe:zbw-zob" facName="zcc" org="-453.375,-294.250" tangent="+42:18:08.00,-080:34:49.00">
      <externalArts localName="rrr" remoteName="rrr"/>
    </hostio>
  </host>
  <etms name="tmu" starsid="lll" device="etms:/dev/ttyS0" rate="19200" />
</sources>
```

NADIN configuration example:

```
<simconfig>

  <comments>Simulated NADIN config</comments>

  <nadin name="czwgzqzn" destination="kzlczqzx" device="nadin:10.226.238.42/10052" />

</simconfig>
```

Appendix F. Messages File Format

General

The messages file is an xml file which defines the messages to be injected during the scenario.

ifmsg

Defines an Inter-Facility message.

Special token substitutions in msg text processed by SDRR:

@@@: remote CID

###: local CID

$hhmm$: time substitution relative to sim starttime

%hhmm%: time substitution relative to current simtime

Required Parameters:

src: Specifies the source facility of the message.

dest: Specifies the destination facility of the message.

time: Specifies the injection time of the message.

(text): Specifies the text of the message (does not have an xml tag).

Optional Parameters:

acid: Specifies the target acid for the message. Any target-specific substitutions will be based on the acid. Messages with an acid will be retried up to 5 times.

literalMode: When enabled, message text will be injected exactly as it appears (no token substitution).

doField23: When set on a TI msg, field23 will be auto-generated based on data in the tracks file.

fdiomsg

Defines a FDIO message.

Required Parameters:

src: Specifies the source facility of the message.

dest: Specifies the destination facility of the message.

time: Specifies the injection time of the message.

(text): Specifies the text of the message (does not have an xml tag).

Optional Parameters:

acid: Specifies the target acid for the message. Any target-specific substitutions will be based on the acid. Messages with an acid will be retried up to 5 times.

literalMode: When enabled, message text will be injected exactly as it appears (no token substitution).

Appendix G. Sample Messages File

```
<?xml version="1.0" encoding="iso-8859-1" ?>

<cmds>

   <ifmsg src="acy" dest="zdc" time="00:00:05.00" ldn="0" >ACY0000000 ::FP USA123 A109/A 1234 300 NOKIY
E$0000$ 210 NOKIY..EPICS..SWANN ::*</ifmsg>

   <ifmsg src="pct" dest="zdc" time="00:04:55.00" ldn="0" acid="FE22203" >PCT0004001 ::DM @@@
$0005$ ::*</ifmsg>

   <ifmsg src="zny" dest="zdc" time="00:05:20.00" ldn="0" acid="N01234" >N000 ::FP N01234 GULF 5500 339
3934/07351 E$0008$ 145 EMJAY..ZIZZI.J174.ORF.J121.CHS ::*</ifmsg>

   <ifmsg src="zny" dest="zdc" time="00:05:23.00" ldn="0" acid="FE25201" >N001 ::TA FE25201 O40
::*</ifmsg>

   <ifmsg src="zny" dest="zdc" time="00:08:20.00" ldn="0" acid="N01234" >N002 ::TI NVF2A N01234/@@@
( 415.625,563.375 )( -215,-262 ) 145 5500 ZCN00 ::*</ifmsg>

   <ifmsg src="nvf" dest="zdc" time="00:08:25.00" ldn="0" acid="N01234" >NVF0008000 ::TA @@@
O1S::*</ifmsg>

   <fdiomsg host="zdc" time="00:10:34.00" device="SH1" ldn="0" ccu="ccu3i" acid="N864TM"
>\x01\x12\x00\x00\x15\x25\x02\x00FP N864TM F2TH/Q 0301 460 IAD P$0016$ 400 IAD/O..DAILY..ACY/O
\x03</fdiomsg>

   <ifmsg src="nvf" dest="zdc" time="00:13:55.00" ldn="0" acid="N01234" >NVF0013001 ::TI W51 @@@ (
-60.750,167.250 )( -223,-255 ) NVF2E ::*</ifmsg>

   <fdiomsg host="zdc" time="00:15:00.00" device="RDU" ldn="0" ccu="ccu1i" acid="USA812"
>\x09\x10\x00\x00\x15\x25\x02\x00FP USA812 MD80/A 0042 420 RDU P$0020$ 270
RDU..ROA..AIR..CLE\x03</fdiomsg>

   <ifmsg src="pct" dest="zdc" time="00:15:29.00" ldn="0" acid="N864TM" >PCT0015002 ::DM @@@ $0016$
::*</ifmsg>

</cmds>
```

Appendix H. Response Control File Format

General

The response control file is an xml file which customizes SDRR behavior in response to real-time simulated events.

response

Defines a custom message response.

Required Parameters:

facility: Specifies the facility that will be affected by the response control statement.

acid: Specifies the target acid for the statement.

mode: Specifies the message response for SDRR to take. One of the following must be specified: “NO_REPLY”, “DX”, “DR”, “LRM”, or “LAM”.

Optional Parameters:

time: Specifies the start time of the statement. Default is 0.

msgtype: Specifies the message type for which this response control is to be used. Examples are “FP”, “TI”, etc. Default is all message types.

cnt: Specifies the number of times this response control will be active. Specify -1 for “forever”. 0 is invalid. Default is 1.

rejcode: Specifies an optional code to be included in the DR message generated by SDRR. (Only used when **mode**=”DR”).

taCtl

Defines a customized TA response time.

Required Parameters:

facility: Specifies the facility that will be affected by the response control statement.

acid: Specifies the target acid for the statement.

Optional Parameters:

delta: Specifies the time in seconds for an auto-TA response. 0 means immediate. -1 means never. Default is 0 (immediate).

relayCtl

Defines custom relay (HNH) behavior when a TI is received with an “00” in field 16.

Required Parameters:

facility: Specifies the facility that will be affected by the response control statement.

acid: Specifies the target acid for the statement.

Optional Parameters:

At least 1 of these must be specified:

destFac: Specifies the facility to relay this message to.

initialController: The controller to be specified in field 71 of the DA to the TI (SDRR defaults to “22”).

finalController: Used for field 48 of the TA (SDRR defaults to “22”).

vfrCtl

Customizes the FP sent in response to a received VFR FP.

Required Parameters:

facility: Specifies the facility that will be affected by the response control statement.

acid: Specifies the target acid for the statement.

Optional Parameters:

At least one of the following must be specified:

coordFix: Specifies field 6 of the return FP. (SDRR defaults to using the last fix of field 10 in the VFR FP).

bcn: Specifies field 4 of the return FP. (SDRR will auto-assign by default).

Appendix I. Sample Response Control File

```
<tcrs>
   <!-- one DX to FP -->
   <response facility="zdc" acid="DPT01" msgType=”FP” mode="DX"  />

  <! – NO response to all msgs forever -->
   <response facility="zdc" time="00:02:19.00" acid="DPT01" mode=”NO_REPLY” cnt="-1" />

   <!-- auto-TA after 5 seconds -->
   <taCtl facility="zdc" time="00:10:00.00" acid="TEST01" delta="5” />

    <!-- do an immediate TA -->
   <taCtl facility="zdc" time="00:10:00.00" acid="TEST02" delta="0” />

   <! – relay a flight to ZOB -->
   <relayCtl facility="zdc" acid="HNH01" destFac=”zob” />

</tcrs>
```

Appendix J. Revision History

November 24, 2008 (Version 2.1.16 Rev. 1)

- Initial Publication

January 28, 2009 (Version 2.2.3 Rev. 2)

- Added diagram in SDRR overview
- Several editorial changes through the text

June 26, 2009 (Version 2.2.6 Rev. 3)

- Added section for surveillance recording
- Added several utilities descriptions (Chapters 6.1 to 6.19)

June 29, 2009 (Version 2.2.6 Rev. 4)

- Updated appendixes A to I

July 6, 2009 (Version 2.2.6 Rev. 5)

- Added Revision History
- Added NADIN configuration example

August 31, 2009 (Version 2.3.1 Rev. 6)

- Added Chapter 6 (Automatic Dependant Surveillance – Broadcast (ADS-B) Support) and modified numbering for the rest of the manual.

January 22, 2010 (Version 2.3.7 Rev. 7)

- Added PIDP facility

May 21, 2010 (Version 2.3.8 Rev. 8)

- Added ExtractCDR Utility
- Added SIRDSplayer Utility
- Added TimeSlice GUI Utility

October 15, 2010 (Version 2.5.1 Rev. 9)

- Updated Appendixes D and E

April 13, 2011 (Version 2.5.11 Rev. 10)

- Added utility TARP