



**SiteShadow**

**Operation and Maintenance Manual**

July 5, 2012

Volume 1: Operation and Maintenance Manual SiteShadow Version 1.1.4  
Revision 1

Copyright © 2010, 2011, 2012

JVN Communications Inc.  
3120 Fire road  
Egg Harbor Township, NJ 08234  
609-569-1477 ext. 202  
[www.jvncommunications.com](http://www.jvncommunications.com)

# Table of Contents

<b>Table of Contents</b> .....	<b>2</b>
<b>1. <u>SiteShadow Overview</u></b> .....	<b>3</b>
<b>2. <u>Getting Started</u></b> .....	<b>4</b>
<b>3. <u>ifshadow</u></b> .....	<b>5</b>
<b>4. <u>relayd</u></b> .....	<b>7</b>
<b>5. <u>avid</u></b> .....	<b>8</b>
<b><u>Appendix A. Revision History</u></b> .....	<b>9</b>

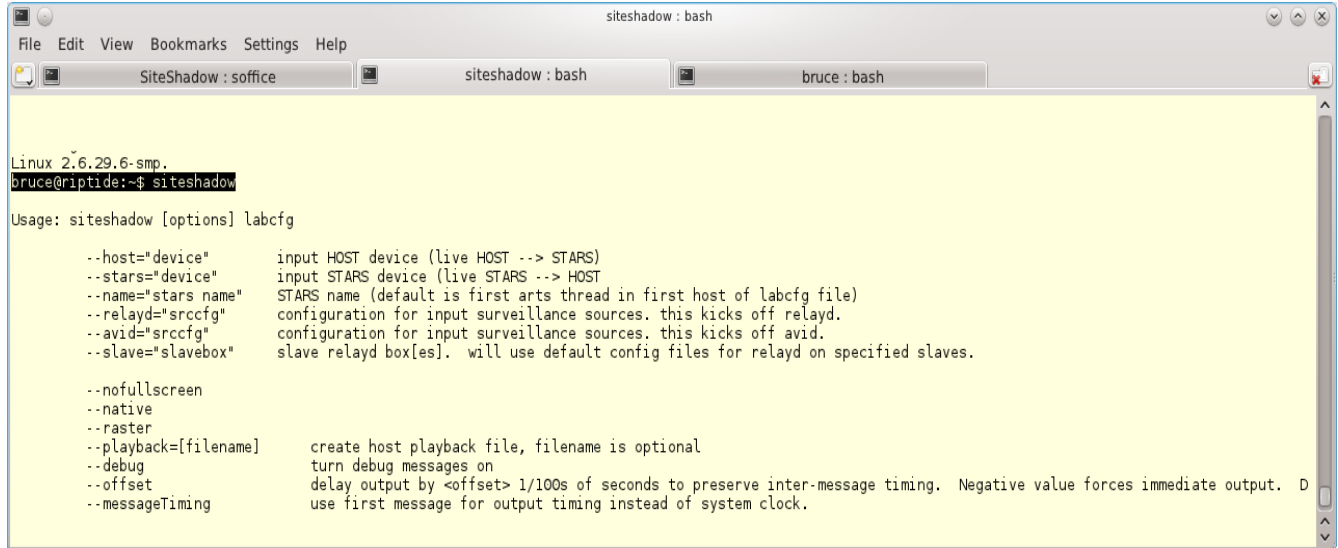
# 1. SiteShadow Overview

The JVN SiteShadow tool executes and manages applications that are needed to sniff live interfacility and surveillance connections while forwarding messages to a “shadowing” system. The applications necessary for this functionality are ifshadow, relayd, and avid.

- ifshadow – interfacility messages
- relayd – surveillance data
- avid – graphical display of relayd message counts

## 2. Getting Started

SiteShadow is launched from the command line of a konsole window. Executing 'siteshadow' with no options will display usage information



```
siteshadow : bash
File Edit View Bookmarks Settings Help
SiteShadow : soffice siteshadow : bash bruce : bash
Linux 2.6.29.6-smp.
bruce@riptide:~$ siteshadow
Usage: siteshadow [options] labcfg

--host="device"      input HOST device (live HOST --> STARS)
--stars="device"    input STARS device (live STARS --> HOST)
--name="stars name" STARS name (default is first arts thread in first host of labcfg file)
--relay="srccfg"    configuration for input surveillance sources. this kicks off relayd.
--avid="srccfg"     configuration for input surveillance sources. this kicks off avid.
--slave="slavebox"  slave relayd box[es]. will use default config files for relayd on specified slaves.

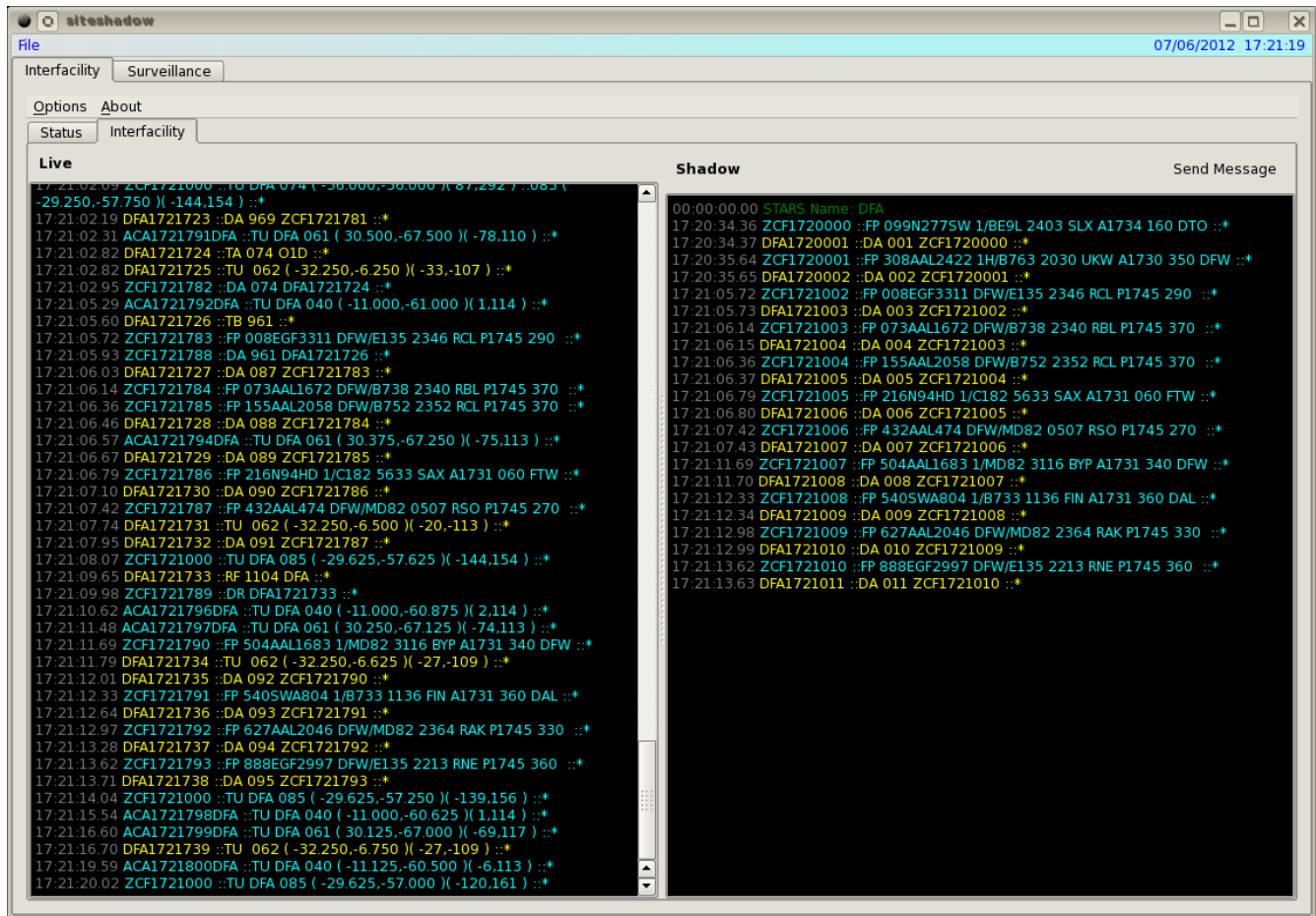
--nofullscreen
--native
--raster
--playback=[filename] create host playback file, filename is optional
--debug            turn debug messages on
--offset           delay output by <offset> 1/100s of seconds to preserve inter-message timing. Negative value forces immediate output.
--messageTiming    use first message for output timing instead of system clock.
```

Sample command line execution

```
siteshadow ./local.xml --relay=./radarSources.xml --host dfw1:/dev/lif2 --stars dfw1:/dev/lif1 --avid=./radarSources.xml
```

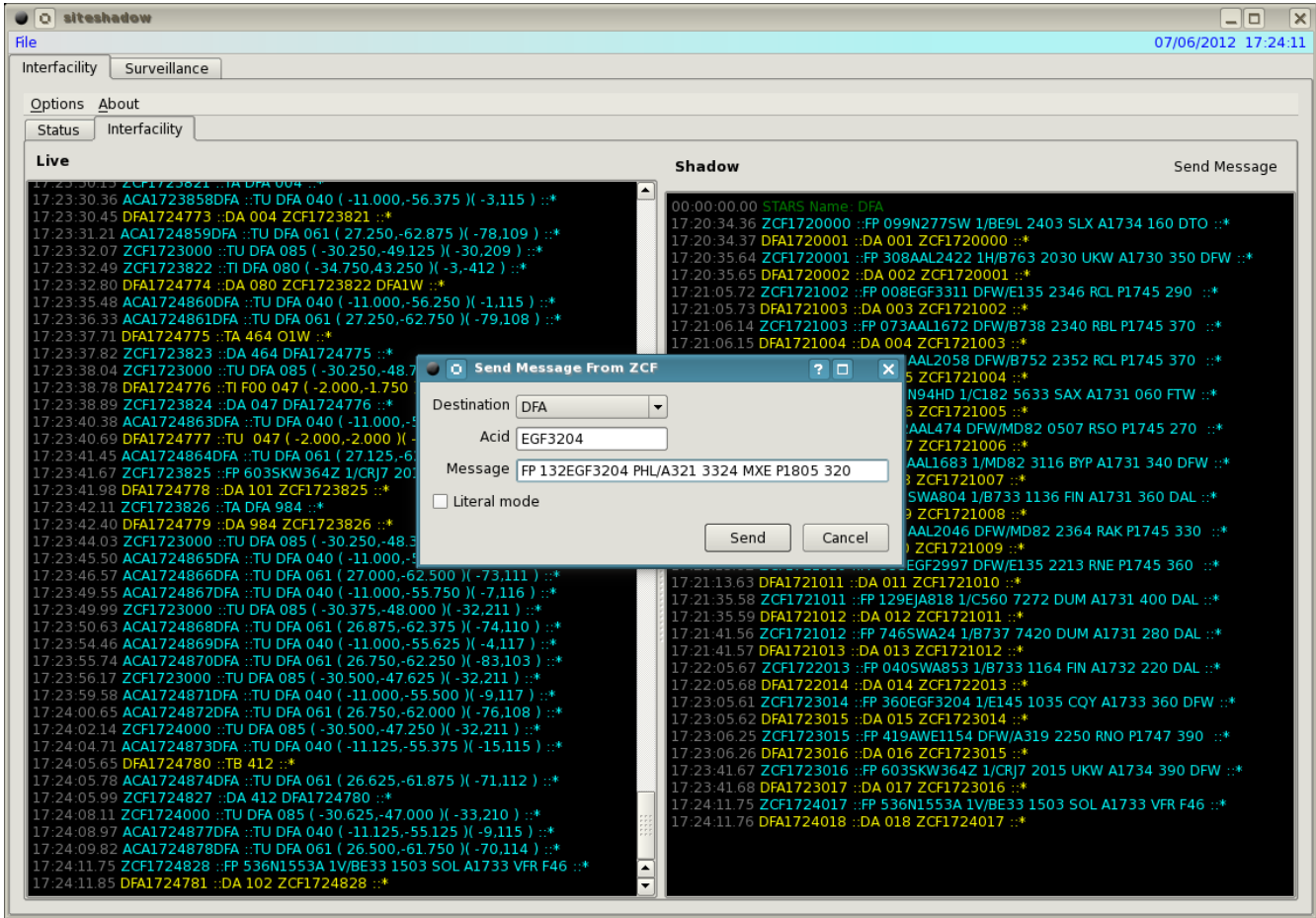
# 3. IFShadow

The JVN IFSHADOW application safely sniffs live IFDT connections with a receive port while passing the IFDT data along a separate bidirectional port to a terminal system/s, (STARS, CARTS,TAMR...). IFSHADOW acts a translator between what it receives from the live IFDT and what it sends to the terminal system/s, converting ECID and TCIDs to match the connected terminal system/s. The terminal system/s actually receives what the real HOST sends and is able to respond back with DA, DR because IFSHADOW is in the middle acting like a smart switch. This allows controller/trainees to actually have real flight plans populate their tab list and auto acquire on tracks.



IFSHADOW includes the ability to pass hand off messages to and from the Host/Terminal. This allows controller/trainees to receive the actual arrival/over-flight hand offs from the center. They are also able to hand off departures tracks to the simulated center, or even to adjacent facilities that IFSHADOW is simulating. The controller will be able to see the response DA/DR messages as if the real HOST/Terminal responded. This is a big help while testing hand offs to adjacent sites.

IFSHADOW includes a helpful real-time injection tab where users can real-time inject their own flight plans, amendments and such to their terminal system. This functionality has proven to be a valuable test feature while debugging IFDT issues. It also allows users to manually send flight plans to tag with stray targets on the glass.



## 4. Relayd

This application is a process that relays JVN formatted messages. The devices used for input and output are defined in configuration files.

Usage : relayd input\_config output\_config [-o <offset>] [-dmst]

-s Use standard input and output config files.

-d Turn debug messages on.

-t Don't fork into background.

-o Delay output by <offset> 1/100s of seconds to preserve inter-message timing. Negative value forces immediate write. Defaults to immediateWrite.

-m Use first message for output timing instead of system clock (for ECGP source only).

Suggested settings : server (no options), client -o (10-100)

### **input configuration:**

```
<root>
```

```
<sources>
```

```
<radar name="dfe" device="/dev/srr0" type="asr9-modes" wx="0" magdev="6.00"  
parrots="1273,1274" scantime="4.75" showDataBlocks="0" elev="676.00" historyAge="0"  
visible="0" pos="+32:52:36.92,-097:00:53.21" color="#dc143c" src="0" historyInterval="0"/>
```

```
</sources>
```

```
</root>
```

### **output configuration:**

```
<root>
```

```
<sources>
```

```
<radar name="dfe" device="/tmp/dfe.srv" type="asr9-modes" wx="0" magdev="6.00"  
parrots="1273,1274" scantime="4.75" showDataBlocks="0" elev="676.00" historyAge="0"  
visible="0" pos="+32:52:36.92,-097:00:53.21" color="#dc143c" src="0" historyInterval="0"/>
```

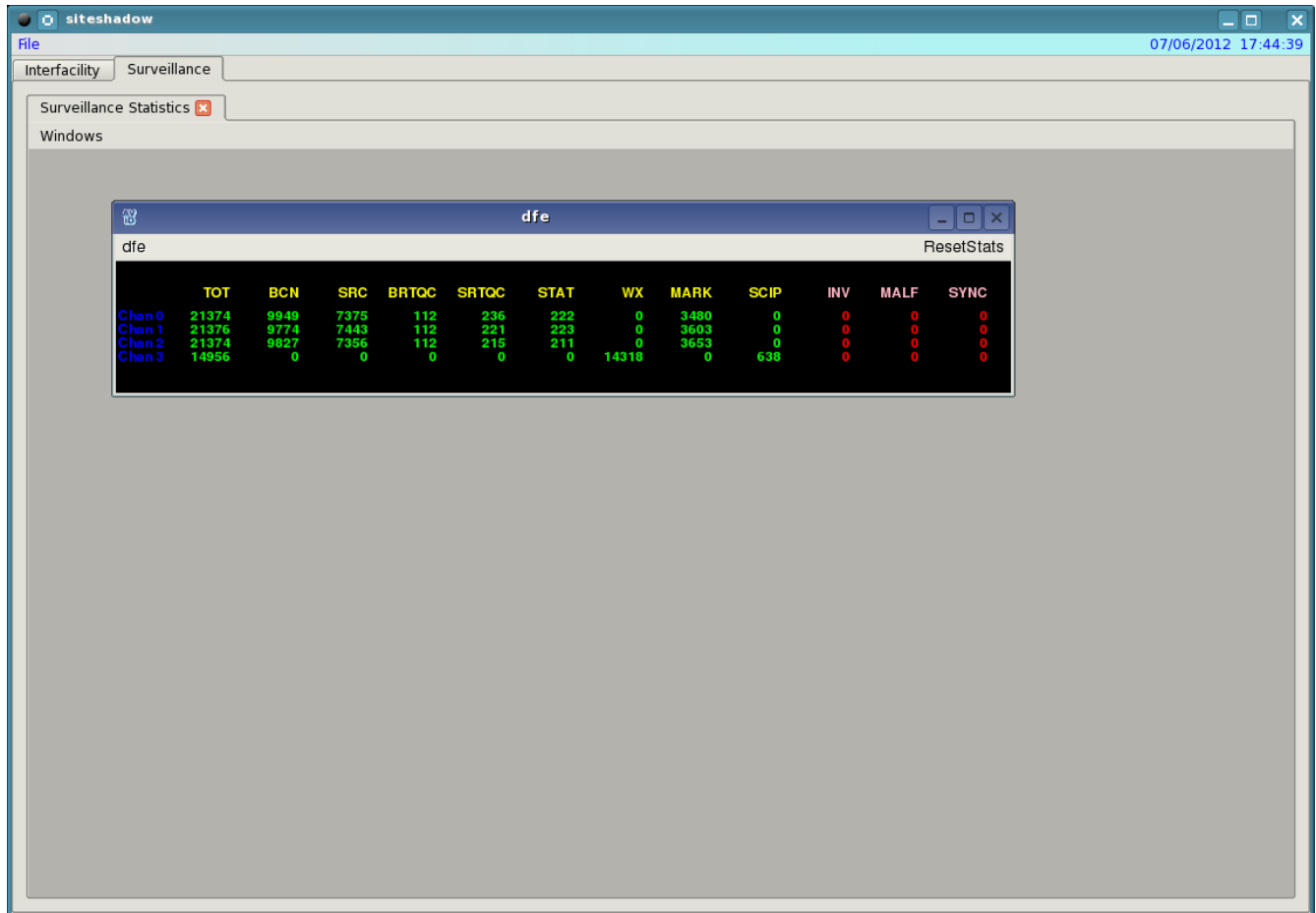
```
</sources>
```

```
</root>
```

Using relayd with these configurations will cause any data from /dev/srr0 to be written to a file /tmp/dfe.srv

# 5. AViD

The Airspace Visualization Display (AViD) displays a graphical representation of messages counts from the relayd process.





# Appendix A. Revision History

July 5, 2012 (Version 1.1.4 Rev. 1)

- Initial Publication