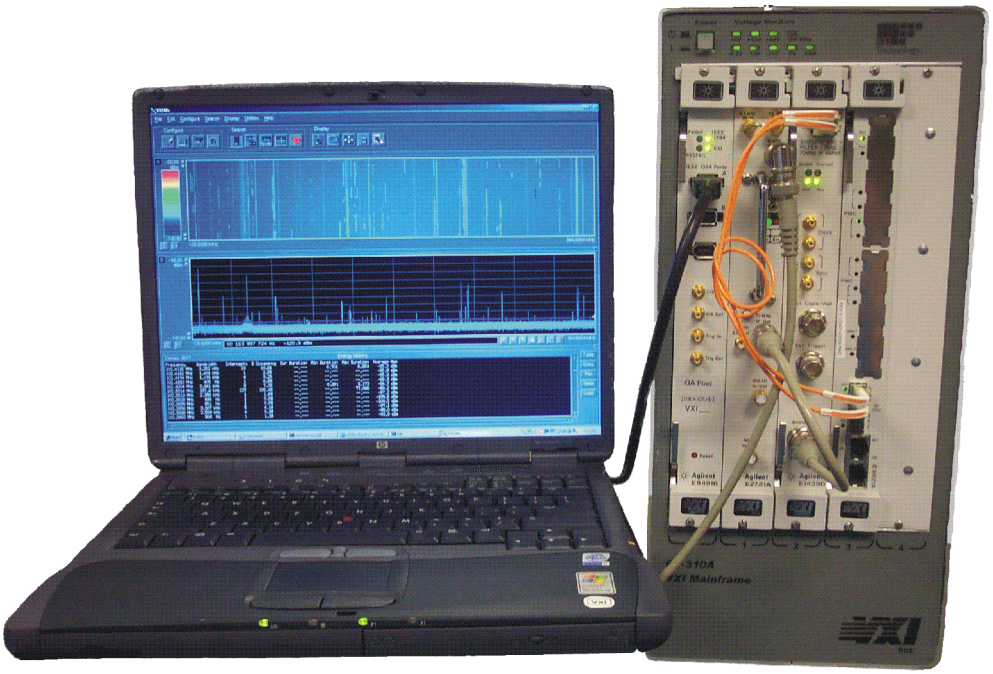
E3238S

Installation and Configuration Reference



Part Number: E3238-90010

Software Version: E3.2

Printed in U.S.A.
Print Date: July 2009

1615 75th Street SW Everett, Washington 98203-6261 U.S.A

NOTICE

The information contained in this manual is subject to change without notice.

AGILENT TECHNOLOGIES MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MANUAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Agilent Technologies shall not be liable for errors contained herein or direct, indirect, special, incidental, or consequential damages in connection with the furnishing, performance, or use of the material.

Trademarks

Windows®,Windows® 2000, and Windows XP are U.S. registered trademarks of Microsoft Corporation in the United States and other countries.

U.S. GOVERNMENT RESTRICTED RIGHTS

Software and technical data rights granted to the federal government include only those rights customarily provided to end user customers. Agilent provides this customary commercial license in Software and technical data pursuant to FAR 12.211 (Technical Data) and 12.212 (Computer Software) and, for the Department of Defense, DFARS 252.227-7015 (Technical Data - Commercial Items) and DFARS 227.7202-3 (Rights in Commercial Computer Software or Computer Software Documentation).

Made in USA

Use of this manual and compact disk(s) or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs can be made for security and back-up purposes only. Resale of the software in its present form or with alterations is expressly prohibited.

Table of Contents

Hardware Installation

This chapter describes the installation and configuration of measurement hardware for the E3238S Signals Development System.

This chapter contains the following topics:

Supported Components

The system may contain the following supported components:

- VXI Mainframe with option 918 (backplane shield kit)
- LTPC2 laptop controller
- E8491B VXI firewire interface
- E1437A[1], E1438A/B/D, E1439A/B/D, or N6830 VXI ADC
- E9821A Signal Processor
- E1472A VXI RF Multiplexer
- E1368/69A VXI Microwave Switch
- E9830A Snapshot and Delay Memory[2]
- bc350/357VXI-C Time and Frequency Processor (IRIG)
- 89431A Tuner (2 - 2650 MHz)[3]
- WJ 9119-1 VXI Tuner (0.5 - 32 MHz, 8 MHz IF BW) used with the E1437 ADC
- E2730/31A/B VXI Tuner (20 MHz - 2.7/6.0 GHz) used with the N6830 and E1439 ADC
- CS-5040 VXI Tuner (0.5 - 20/40/60 GHz)
- DRS SI-9250-2 VME Downconverter (20 MHz - 18 GHz)
- DRS SI-9136B UHF/VHF Digital VME Tuner
- E444xA PSA-Series Spectrum Analyzer (must support option H70 or HY7)
- N6841A RF Sensor
- Conduant LTX2 (E3238S-050) 5.12 TB 1U Disk Array
- Conduant LTX2-35 (E3238S-051) 16.0 TB 2U Disk Array

Notes

Most of the E3238S systems are integrated at the factory by Agilent Technologies. This note describes the recommended system configurations.

For N6841A RF Sensor hardware and software installation, refer to the documentation that came with the N6841A RF Sensor..

EMC filler panels *must* be installed in all empty VXI chassis slots. This is necessary for adequate airflow and provides shielding required to meet EMC regulatory requirements. Part number: E8400-60202.

To setup, the only installation steps you should have to perform are to connect the cables. See Typical System Configurations (cabling) on page 13 and Fiber Optic Cabling of the DSP Module on page 28.

[1]The E1437A is obsolete. Information provided in this manual is for the continued use of previously-purchased units.

[2]The E9830A is obsolete. Information provided in this manual is for the continued use of previously-purchased units.

[3]The 89431A is obsolete. Information provided in this manual is for the continued use of previously-purchased units.

Safety Notices

This product has been designed and tested in accordance with accepted industry standards, and has been supplied in a safe condition. The documentation contains information and warnings that must be followed by the user to ensure safe operation and to maintain the product in a safe condition.

For Safety and Regulatory information, please refer to the General Specifications on page 231 and Safety Summary on page 232 of this manual.

Hardware Installation Process

Note Prior to System Installation, make sure you thoroughly read the safety notices found in the Specifications section - Page 197.

Installing new modules, replacing modules, or reconfiguring the system requires two steps:

1. Install the module(s): set address switches and connect cables.

   Switch settings:

   - E2730B/E2731B VHF/UHF Tuner (page 48)
   - SI-9250 Block Downconverter (page 49)
   - SI-9136B VHF/UHF Digital VME Tuner (page 50)
   - Communication Solutions CS-5040 Microwave Tuner (page 52)
   - E1437A, E1438A/B/D, E1439A/B/D, N6830 VXI ADC (page 39)
   - E9821A VXI Signal Processing Module (page 42)

   Cabling information begins on page 13.

2. Edit the configuration file (`e3238s.cfg`) so that it correctly defines the hardware installed.

All the commands that define hardware configuration are listed in the Hardware Configuration Reference on page 89.

Note The E3238S passes IEC/EN 61326-1 when the front cover and ground wire are installed per the EMI and Cable Protection Kit Installation Note that was included in your shipment. If you would like to upgrade a 5-slot (MFRAME1) or 13-slot (E8403A or E8404A) VXI mainframe that was shipped prior to April 1, 2009, contact Agilent Technologies. There is no upgrade for 6-slot VXI mainframes shipped prior to May 1, 2009.

About Tuners and ADCs

ADC Only If the baseband frequency range of the ADC is sufficient, no tuner is required. Only the N6830/HF ADC is recommended for use without a tuner.

WJ9119-1 A pair of VXI modules (LO and RF tuner) that covers 2 MHz to 32 MHz. It is designed to work with the E1437A ADC.

E2730/31B A single-slot VHF/UHF VXI tuner designed to be used with the N6830 ADC and the E1439D ADC.

E444xA PSA This is a spectrum signal analyzer with a tuner section that is used by the system.

SI-9250 This is a microwave block downconverter that can be added to a VHF/UHF configuration.

SI-9136B This is a Dual Channel VHF/UHF VME Tuner.

CS5040 This is a microwave VXI tuner.

89431A A non-VXI tuner controlled by RS-232 interface connected to the E9821A DSP.

N6841A A small RF signal monitoring device housed in a weatherproof enclosure.

Tuner-ADC Combinations

Tuner driver names	E1437A F_1 - F_2 (MHz) IF BW (MHz)	E1438 F_1 - F_2 (MHz) IF BW (MHz)	E1439/70 F_1 - F_2 (MHz) IF BW (MHz)	E1439/BB[1] F_1 - F_2 (MHz) IF BW (MHz)	N6830/HF F_1 - F_2 (MHz) IF BW (MHz)	N6830/70 F_1 - F_2 (MHz) IF BW (MHz)
ADC Only	0 - 8 8	0 - 40 40	52 - 88 36	0 - 37.109375 37.109375	**.1 - 32** **32, 16, or 8**	52 - 88 36
WJ9119-1	**1 - 32** **6.88**	1 - 32 8	NS	1.0 - 32 8	NS	NS
E2730B	NS	NS	**20-2,700** **36**	NS	NS	**20-2,700** **36**
E2731B	NS	NS	**20 - 6,000** **36**	NS	NS	**20 - 6,000** **36**
E444xA PSA + HY7[2]	NS	NS	**.1 - 26500** **36**	NS	NS	**.1 - 26500** **36**
SI-9250[3]	NS	NS	**20 - 18000** **36**	NS	NS	**20 - 18000** **36**
SI-9136B	NS	NS	**5 - 3000** **30**	NS	NS	**5 - 3000** **30**
CS5040	NS	NS	**500 - 20,000[4]** **36**	NS	NS	**500 - 20,000[5]** **36**
HP89431A	2 - 2,650 6.34	2 - 2,650 8.34375	NS	2 - 2,650 8.34375	NS	NS

NS = the combination is *not supported*
Bold numbers indicate optimal combinations

1. In baseband mode the E1439 has no input attenuation and its fullscale input level is -21 dBm. Since the output level for most of these tuners is -6 dBm, an external attenuator should be used to avoid overloading the ADC input.
2. Option H70 is also supported but HY7 has better performance.
3. Requires either the E2730B or the E2731B tuner.
4. **The range may be extended to >100 GHz by connecting the appropriate block down converter to the CS5040 VXI module.**
5. **The range may be extended to >100 GHz by connecting the appropriate block down converter to the CS5040 VXI module.**

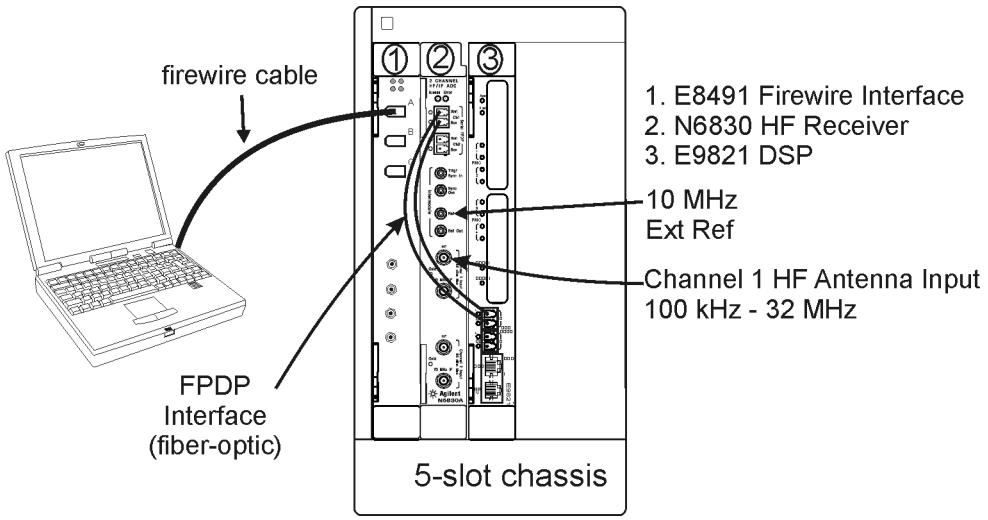
Typical System Configurations (cabling)

The following figures show examples of module and cable placement for various configurations. Individual module configurations are described in detail starting on page 38.

HF Search and Collection (N6830)

Figure 1 shows an N6830 HF receiver configuration in a 5-slot chassis (MFRAME1) that provides HF search and collection in a small package.

Figure 1. HF search and collection



Data is passed to the E9821 DSP via the fiber-optic FPDP (front-panel data port) interface. See Fiber Optic Cabling of the DSP Module on page 28 for more information.

Depending on the PMC modules installed on the E9821 DSP board, this system can perform search and provide 32 channels of narrowband channelization for collection. At least one 32-channel digital downconverter (DDC, option 200) is required for narrowband channelization. See DSP Configuration on page 36. Additional E9821 DSP boards can be added to the system to provide additional narrowband channels.

Note

The typical E9821 DSP configuration for most signal types is one dual G4 processor (option 101) for each 32 channel digital downconverter (DDC, option 200). With this configuration, each additional E9821 DSP module adds 64 digital downconverters for narrowband channelization.

The block diagram for this configuration appears in figure 22 on page 35.

Installation of the controller and firewire interface is discussed on page 37.

Setting address switches:

- N6830 address settings appear on page 39.
- Switch settings for the E9821 are on page 42.

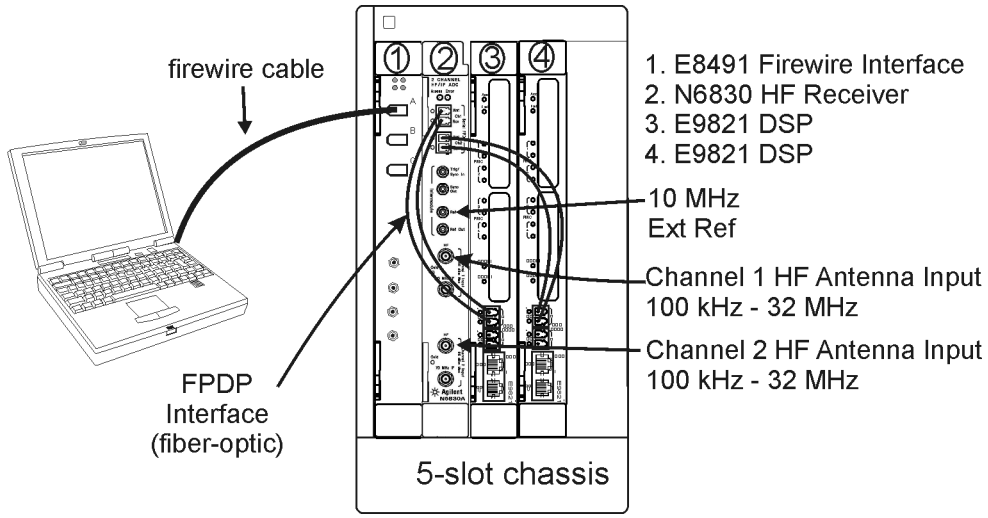
Note

Because of the heat generated by the DSPs, the FAN SPEED switch on the back of the 5-slot and the E1421B 6-slot mainframes should always be set to HIGH.

Dual Channel HF Search and Collection (N6830)

Figure 2 shows a configuration that provides two independent N6830 HF search and collection systems in a 5-slot chassis (MFRAME1).

Figure 2. HF search and collection



Note The two HF search and collection systems are totally independent of each other. In this configuration, two instances of the E3238S application run in the same computer, so only one software license is required.

Data is passed to the E9821 DSP via the fiber-optic FPDP (front-panel data port) interface. See Fiber Optic Cabling of the DSP Module on page 28 for more information.

Note The external reference input is used by both channels. When using the external reference input, both e3238s.cfg files must have the same setting for the searchRx1.adcClock parameter (External). The E3238S application that is started last will always override the searchRx1.adcClock parameter setting of the E3238S application started first.

Depending on the PMC modules installed on the E9821 DSP boards, both instances of the E3238S can perform search and provide 32 channels of narrowband channelization for collection. At least one 32-channel digital downconverter (DDC, option 200) is required for narrowband channelization. See DSP Configuration on page 36. Additional E9821 DSP boards can be added to the system to provide additional narrowband channels.

Note The typical E9821 DSP configuration for most signal types is one dual G4 processor (option 101) for each 32 channel digital downconverter (DDC, option 200). With this configuration, each additional E9821 DSP module adds 64 digital downconverters for narrowband channelization.

The block diagram for this configuration appears in figure 22 on page 35.

Installation of the controller and firewire interface is discussed on page 37.

Setting address switches:

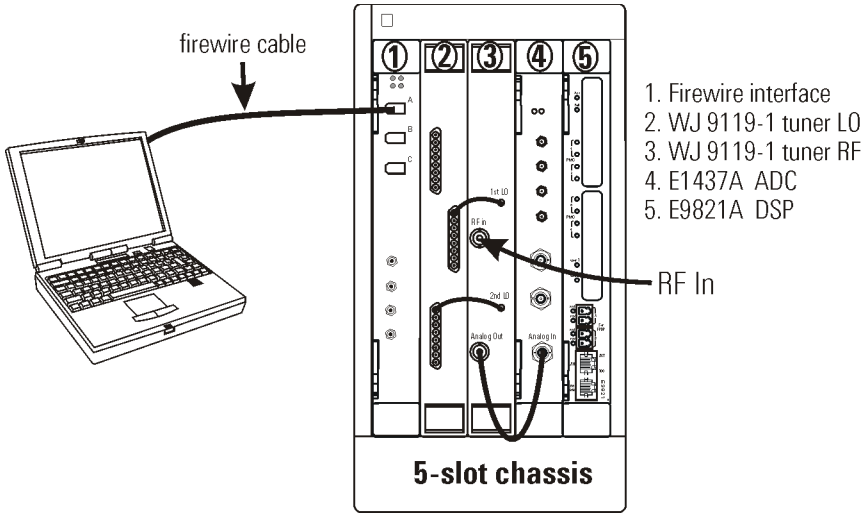
- N6830 address settings appear on page 39.
- Switch settings for the E9821 are on page 42.

Note Because of the heat generated by the DSPs, the FAN SPEED switch on the back of the 5-slot and the E1421B 6-slot mainframes should always be set to HIGH.

HF Search and Collection (E1437)

Figure 3 shows an E1437 ADC configuration that provides HF search and collection in a 5 slot VXI mainframe (MFRAME1).

Figure 3. HF search and collection



Depending on the PMC modules installed on the E9821A board, this system can perform search and provide 32 channels of narrowband channelization for collection. One 32-channel DDC (option 200) is required for narrowband channelization. See DSP Configuration on page 36. E9821 DSP modules with option 200 can be added to 6 slot or 13 slot systems for additional narrowband channels.

The block diagram for this configuration appears in figure 22 on page 35.

Installation of the controller and firewire interface is discussed on page 37.

Setting address switches:

- ADC address settings appear on page 39.
- Switch settings for the E9821 are on page 42.

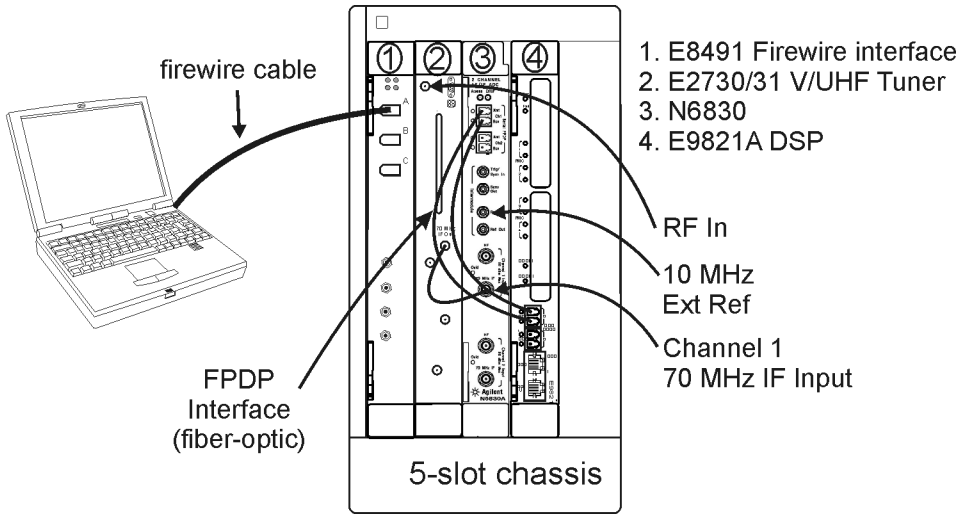
Note

Because of the heat generated by the DSPs, the FAN SPEED switch on the back of the 5-slot and the E1421B 6-slot mainframes should always be set to HIGH.

VHF/UHF Search and Collection (N6830)

Figure 4 shows a configuration in a 5-slot chassis (MFRAME1) that provides V/UHF search and collection in a small package.

Figure 4. VHF/UHF search and collection



Data is passed to the E9821 DSP via the fiber-optic FPDP (front-panel data port) interface. See Fiber Optic Cabling of the DSP Module on page 28 for more information.

Depending on the PMC modules installed on the E9821 board, this system can perform search and provide 32 channels of narrowband channelization. At least one 32-channel DDC (option 200) is required for channelization. See DSP Configuration on page 36. An additional E9821 DSP board with three 32 channel DDCs can be added to the 5 slot system to provide 128 narrowband channels in a 5 slot mainframe. Additional narrowband channels can be added by using a 6 slot or 13 slot mainframe and adding more E9821 DSP modules.

Note

The typical E9821 DSP configuration for most signal types is one dual G4 processor (option 101) for each 32 channel digital downconverter (DDC, option 200). With this configuration, each additional E9821 DSP adds 64 digital downconverters for narrowband channelization.

The block diagram for this configuration appears in figure 22 on page 35.

Installation of the controller and firewire interface is discussed on page 37.

Setting address switches:

- E2730/E2731 (VHF/UHF) tuner configuration information is on page 48.
- N6830 address settings appear on page 39.
- Switch settings for the E9821A are on page 42.

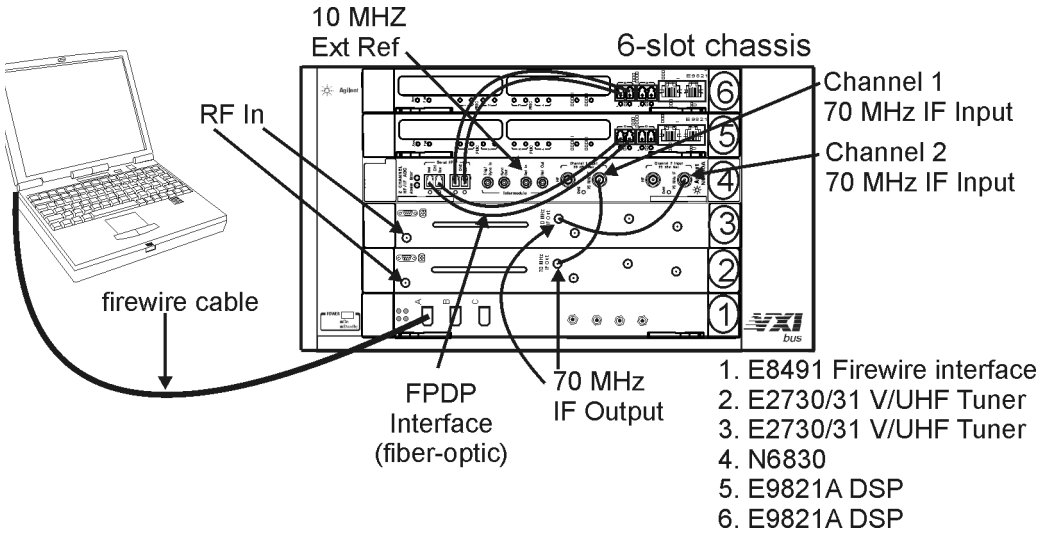
Note

Because of the heat generated by the DSPs, the FAN SPEED switch on the back of the 5-slot and the E1421B 6-slot mainframes should always be set to HIGH.

Dual Channel VHF/UHF Search and Collection (N6830)

Figure 5 shows a configuration in a 6-slot VXI mainframe that provides two independent VHF/UHF search and collection systems in a small package.

Figure 5. VHF/UHF search and collection



Note The two VHF/UHF search and collection systems are totally independent of each other. In this configuration, two instances of the E3238S application run in the same computer, so only one software license is required.

Data is passed to the E9821 DSP via the fiber-optic FPDP (front-panel data port) interface. See Fiber Optic Cabling of the DSP Module on page 28 for more information.

Note The external reference input is used by both channels. When using the external reference input, both e3238s.cfg files must have the same setting for the searchRx1.adcClock parameter (External). The E3238S application that is started last will always override the searchRx1.adcClock parameter setting of the E3238S application started first.

Depending on the PMC modules installed on the E9821 DSP boards, both instances of the E3238S can perform search and provide 32 channels of narrowband channelization for collection. At least one 32-channel digital downconverter (DDC, option 200) is required for narrowband channelization. See DSP Configuration on page 36. Additional E9821 DSP boards can be added to 13 slot systems to provide additional narrowband channels.

Note The typical E9821 DSP configuration for most signal types is one dual G4 processor (option 101) for each 32 channel digital downconverter (DDC, option 200). With this configuration, each additional E9821 DSP adds 64 digital downconverters for narrowband channelization.

The block diagram for this configuration appears in figure 22 on page 35.

Installation of the controller and firewire interface is discussed on page 37.

Setting address switches:

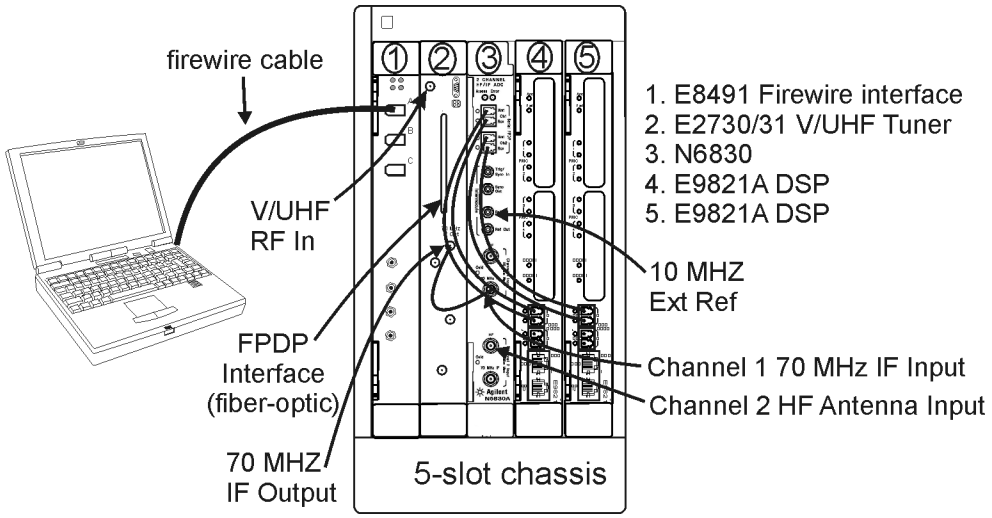
- E2730/E2731 (VHF/UHF) tuner configuration information is on page 48.
- N6830 address settings appear on page 39.
- Switch settings for the E9821A are on page 42.

Note Because of the heat generated by the DSPs, the FAN SPEED switch on the back of the 5-slot and the E1421B 6-slot mainframes should always be set to HIGH.

Simultaneous HF/VHF/UHF Search and Collection (N6830)

Figure 6 shows a configuration that provides an HF search and collection system and a VHF/UHF search and collection system in a 5 slot mainframe (MFRAME1).

Figure 6. HF/VHF/UHF search and collection



Data is passed to the E9821A DSP via the fiber-optic FPDP (front-panel data port) interface. See Fiber Optic Cabling of the DSP Module on page 28 for more information.

Note

The external reference input is used by both channels. When using the external reference input, both e3238s.cfg files must have the same setting for the searchRx1.adcClock parameter (External). The E3238S application that is started last will always override the searchRx1.adcClock parameter setting of the E3238S application started first.

Depending on the PMC modules installed on the E9821A board, this system can perform search and provide 32 channels of narrowband channelization for HF and 32 channels of narrowband channelization for V/UHF. At least one 32-channel DDC (option 200) is required in each E9821for channelization. See DSP Configuration on page 36. Additional E9821 DSP boards can be added to 6 or 13 slot VXI mainframes to provide additional narrowband channels.

Note

The HF search and collection system and the V/UHF search and collection system are totally independent of each other. In this configuration, two instances of the E3238S application run in the same computer, so only one software license is required.

The block diagram for this configuration appears in figure 22 on page 35.

Installation of the controller and firewire interface is discussed on page 37.

Setting address switches:

- E2730/E2731 (VHF/UHF) tuner configuration information is on page 48.
- N6830 address settings appear on page 39.
- Switch settings for the E9821A are on page 42.

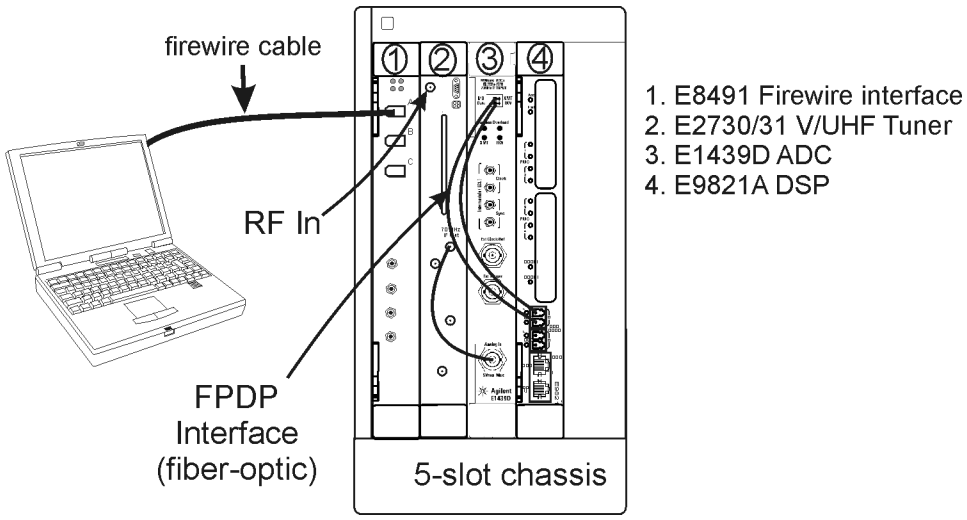
Note

Because of the heat generated by the DSPs, the FAN SPEED switch on the back of the 5-slot and the E1421B 6-slot mainframes should always be set to HIGH.

VHF/UHF Search and Collection (E1439D)

Figure 7 shows a configuration in a 5-slot chassis (MFRAME1) that provides V/UHF search and collection in a small package.

Figure 7. VHF/UHF search and collection



When the ADC is an E1439B/D the data is passed to the E9821A via the fiber-optic FPDP (front-panel data port) interface instead of the backplane local bus. See Fiber Optic Cabling of the DSP Module on page 28 for more information.

Depending on the PMC modules installed on the E9821A board, this system can perform search and provide 32 channels of narrowband channelization. At least one 32-channel DDC (option 200) is required for narrowband channelization. See DSP Configuration on page 36. An additional E9821 DSP board can be added to provide a maximum of 128 narrowband channels in a 5 slot mainframe. Additional E9821 DSP boards can be added to 6 or 13 slot VXI mainframes to provide additional narrowband channels.

Note

The typical E9821 DSP configuration for most signal types is one dual G4 processor (option 101) for each 32 channel digital downconverter (DDC, option 200). With this configuration, each additional E9821 DSP adds 64 digital downconverters for narrowband channelization.

The block diagram for this configuration appears in figure 22 on page 35.

Installation of the controller and firewire interface is discussed on page 37.

Setting address switches:

- E2730/E2731 (VHF/UHF) tuner configuration information is on page 48.
- ADC address settings appear on page 39.
- Switch settings for the E9821A are on page 42.

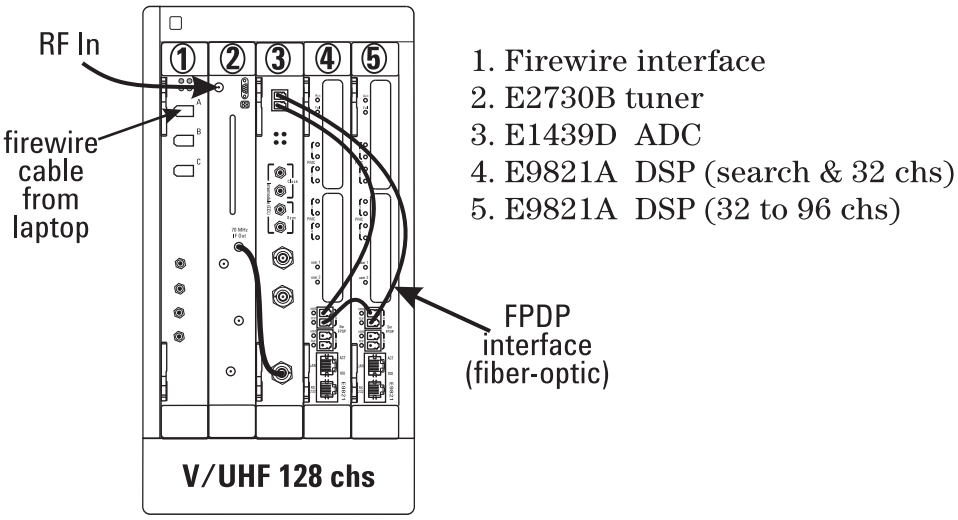
Note

Because of the heat generated by the DSPs, the FAN SPEED switch on the back of the 5-slot and the E1421B 6-slot mainframes should always be set to HIGH.

128 Channel VHF/UHF Search and Collection (E1439D)

Figure 8 shows a 128 channel narrowband V/UHF collection configuration.The block diagram for this configuration appears in figure 22 on page 35.

Figure 8.
VHF/UHF collection



Note

The typical E9821 DSP configuration for most signal types is one dual G4 processor (option 101) for each 32 channel digital downconverter (DDC, option 200). With this configuration, each additional E9821 DSP adds 64 digital downconverters for narrowband channelization. The system above would have a total of 96 narrowband channels.

The block diagram for this configuration appears in figure 22 on page 35.

Installation of the controller and firewire interface is discussed on page 37.

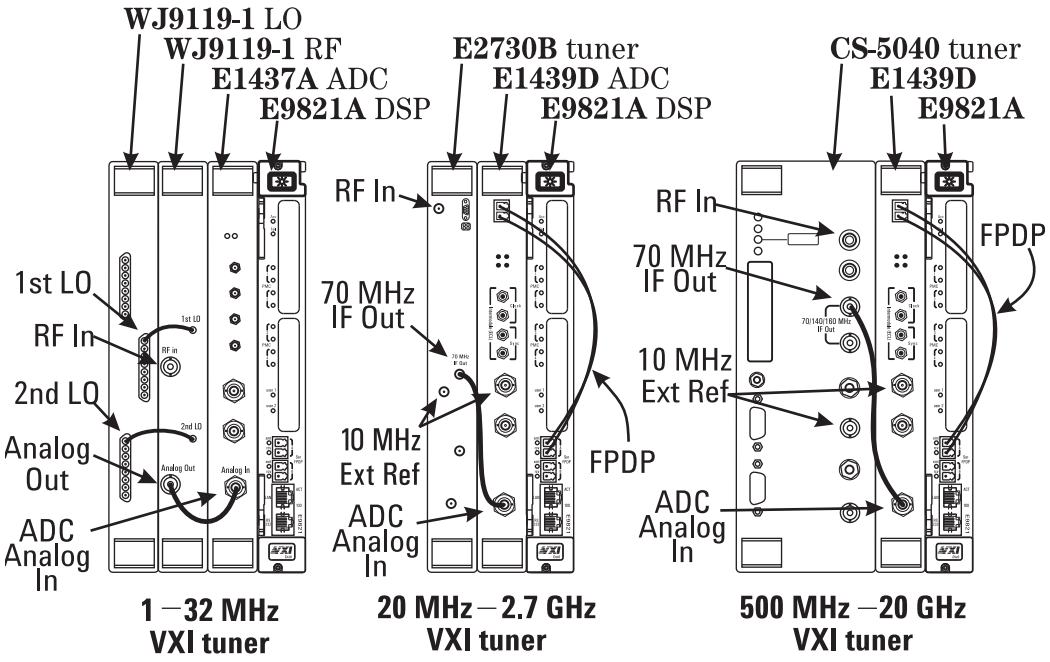
Setting address switches:

- E2730/E2731 (VHF/UHF) tuner configuration information is on page 48.
- ADC address settings appear on page 39.
- Switch settings for the E9821A are on page 42.

VXI Tuner Cabling

Cabling for VXI tuners is shown in figure 9.

Figure 9. VXI tuners: -WJ9119-1 -E2730A -CS5040



Setting address switches:

- WJ-9119 (HF) tuner configuration information is on page 47.
- E2730A (VHF/UHF) tuner configuration information is on page 48.
- CS-5040 (microwave) tuner configuration information appears on page 52.
- ADC address settings appear on page 39.
- Switch settings for the E9821A are on page 42.

The fiber-optic interface between the ADC and the DSP requires two lines and must use port A on the E9821A. See "Fiber Optic Cabling of the DSP Module" on page 28.

The purpose of the figure above is to illustrate cabling. The slots in which the *tuner* modules are installed or order of placement is not critical.

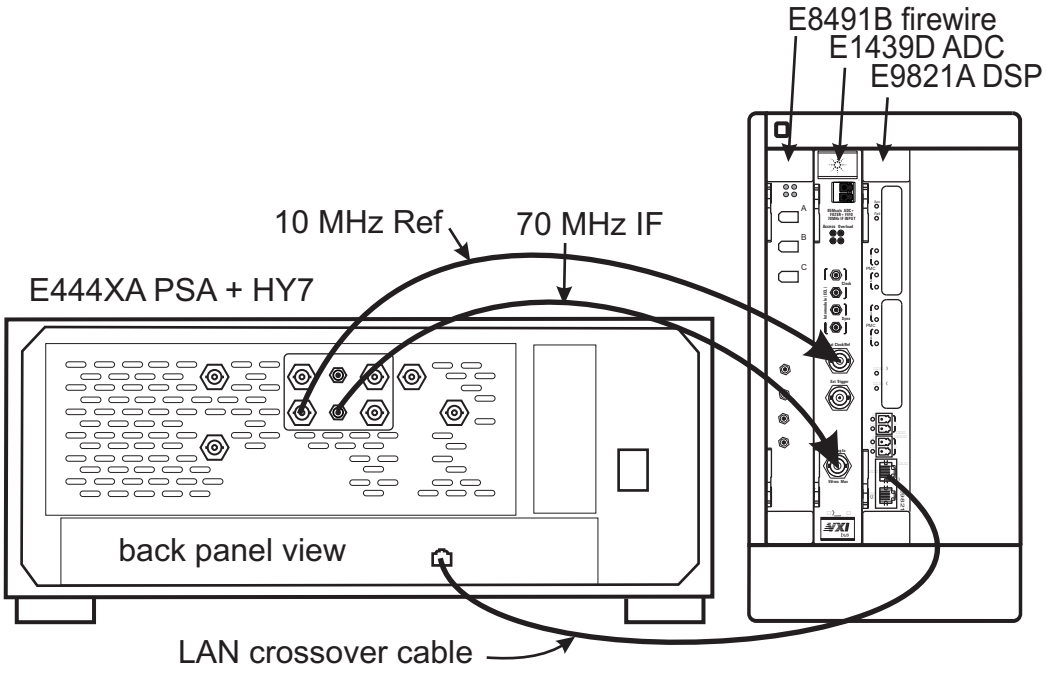
However, when the searchRx.adcDataPort (page 135) resource is **`LocalBus`**, the ADC and DSP modules *must* be installed in adjacent slots with the ADC on the left and the DSP on the right, as shown in figure 9.

See also, the table on Page 11.

PSA Tuner Cabling

The cabling for the E444xA PSA + HY7[1] tuner is shown in figure 10. This uses the tuner section of the PSA and controls it via a LAN connection to the E9821A DSP.

Figure 10.
PSA tuner cabling



A network hub or switch can be used between the LAN ports. Such a configuration allows the use of conventional LAN cables.

When a hub is used between the PSA and the E9821A, its lights should blink to indicate that network traffic is flowing properly.

On the E9821, the link light on the LAN connector illuminates only when connected to a 100 Mbps device. The PSA has a 10 Mbps interface. If a 100 Mbps network hub is used instead of the LAN crossover cable, the E9821A LAN connector light will illuminate.

The configuration file entries for this tuner are as follows:

```
searchRx1.tuner1.tunerModel:    PSA
searchRx1.tuner1.tunerInterfaceParm: 192.168.0.3, 26500, 0, 45, 0, 9000
```

where:

```
192.168.0.3 = LAN IP address or hostname (see next topic)
26500 = Stop Frequency:  3000 to 50000 MHz
    0 = Freq Reference:  0 (Internal) or 1 (External)
   45 = Settling Time:   1 to 1000 mSec Typical: 45 mSec
    0 = Option 1DS preamp: 0=Off, 1=On (+28 dB gain), 110(Option 110 On)
 9000 = IF gain (milli-dB) 9000 (+9dB for HY7), -6000 (for H70)
```

[1]Option H70 is also supported but HY7 has better performance.

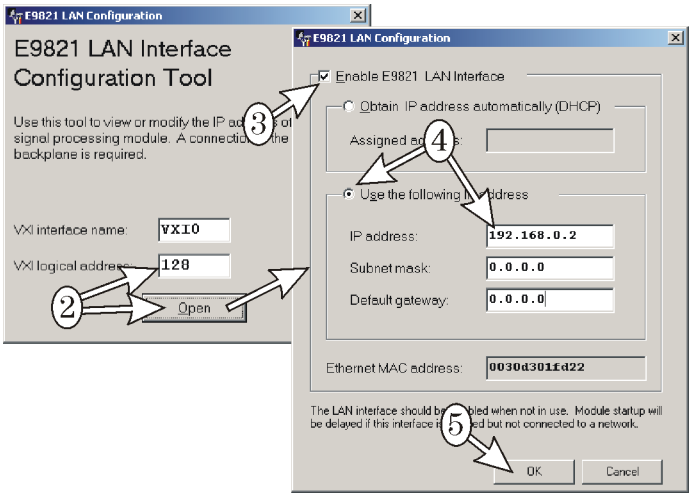
PSA Tuner Interface Setup

E9821A LAN Setup

The E9821A must be configured to work with the PSA as follows:

1. Start the E9821 LAN setup utility:
   Start - Programs - Agilent E3238S - Tools - E9821 LAN Configure
2. Enter the VXI logical address of the E9821A (usually 128) and click Open
3. Enable the E9821 LAN Interface
4. Enable the use of an IP address and enter a unique value such as shown here.
   192.168.0.2
   Enter zeros for the subnet mask and the default gateway as shown.
5. Click OK to finish.

Figure 11. E9821A LAN setup



PSA LAN Setup

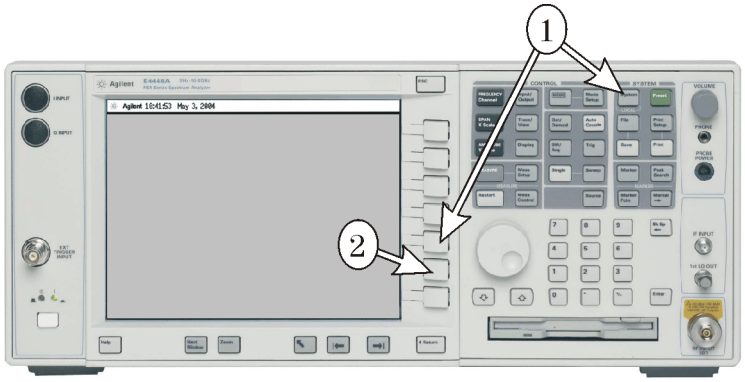
To set the IP address for the PSA tuner:

Note

The IP address for the PSA must be different than the IP address used for the E9821A. The LAN cable provided with the PSA for firmware updates can not be used for this application.

1. Press **System - Config I/O - IP Address** and enter an address like 192.168.0.3
   The default settings for Subnet Mask (255.255.0.0) and Gateway (0.0.0.0) are good for use on a private LAN such as described here.
2. Press the **SCPI LAN** softkey and verify that the **SCPI Socket Server** is **On** and the port is **5025**

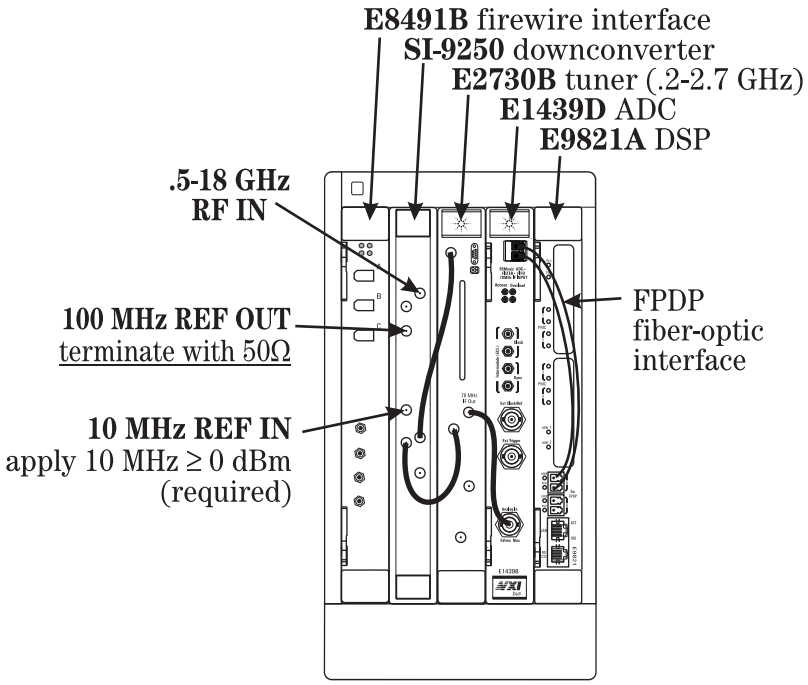
Figure 12. PSA LAN setup



SI-9250 Cabling

Cabling for the SI-9250-2 tuner is shown in figure 13.

Figure 13.
SI-9250 cabling



Special requirements:

- A 50Ω termination must be installed on the 100 MHz Reference output.
- A 10 MHz, 0 dBm signal must be applied to the 10 MHz Reference input

The configuration file entries for this tuner are as follows:

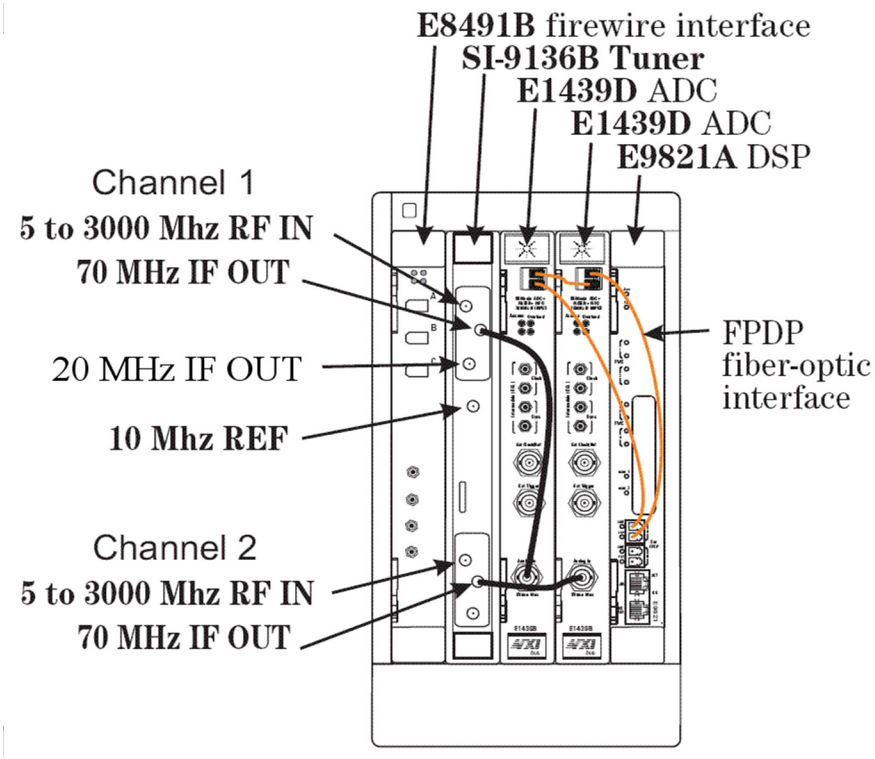
```
searchRx1.tuner1.tunerModel:   SI9250
searchRx1.tuner1.tunerInterfaceParm: 144, 0, 10, 136, 0, 3

where:

144 = logical address of SI-9250
  0 = Reference input: 0 = int or ext 10MHz, 1 = ext 100MHz
 10 = SI-9250 settling time in ms
136 = logical address of companion tuner, E2730/31B
  0 = 10MHz Ref input: 0 = internal, 1 = external
  3 = E273x settling time in milliseconds, typically 3 to 5 ms
```

For directions on setting the address, see SI-9250 Block Downconverter (page 49).

SI-9136B Cabling Dual search receiver configuration

Cabling for the SI-9136B tuner, when used in a dual search receiver configuration is shown in figure 14.

Figure 14. SI-9136B Dual search receiver cabling



Special requirements:

- The SI-9136B requires a VME/VXI carrier card for installation.

The configuration file entries for this tuner are as follows:

```
searchRx1.tuner1.tunerModel:    SI9136
searchRx1.tuner1.tunerInterfaceParm: 4096, 1, 0, 0, 5, 3, 0, 400
searchRx2.tuner1.tunerModel:    SI9136
searchRx2.tuner1.tunerInterfaceParm: 4096, 2, 0, 0, 5, 3, 0, 400
```

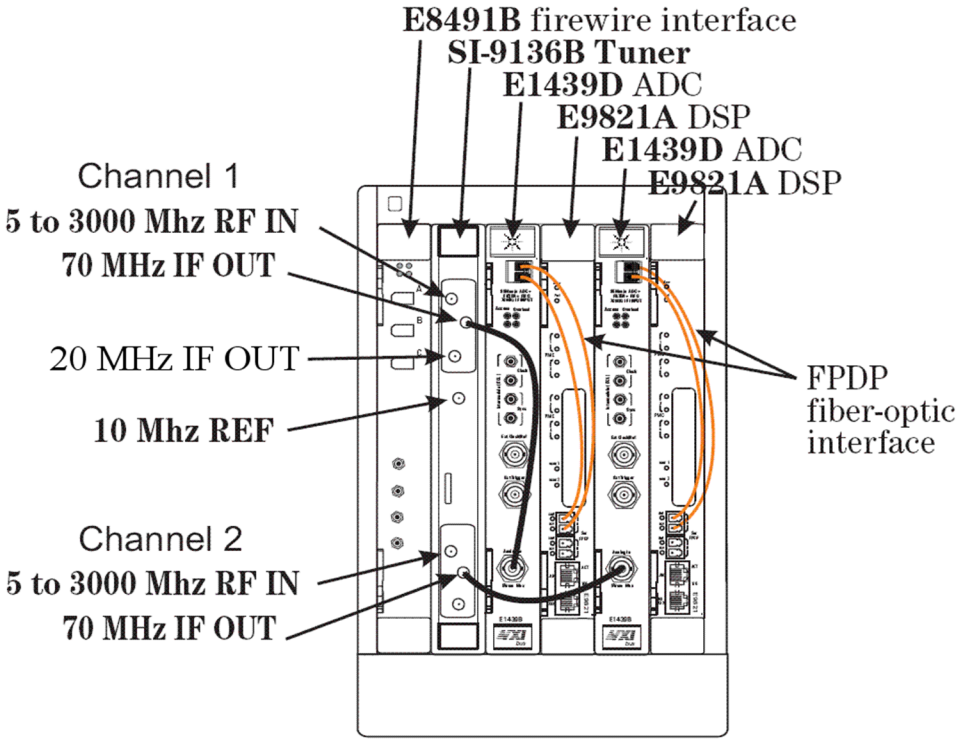
```
where:

0 to 48896 in steps of 2048. Typical: 4096 VME Bus address
1 or 2 = Channel
0 (Independent), 1 (External), 2 (Slave) = LO Mode
0 (Internal), 1 (External) = 10 MHZ Reference
2 MHz to 20 MHz Default: 5 MHz = Start Frequency
0 (Bypass), 1 (Enable) = Preselector Filter
100 To 10000 uSec (400 Typical) = Settling
```

For directions on setting the address, see SI-9136B VHF/UHF Digital VME Tuner (page 50).

SI-9136B Cabling Dual Band configuration

Cabling for the SI-9136B tuner, when used in a dual band configuration is shown in figure 15.[1]

Figure 15. SI-9136B Dual Band cabling



Special requirements:

- The SI-9136B requires a VME/VXI carrier card for installation.
- 2 separate instances of the E3238S software with separate config files

The first configuration file entries for tuner #1 are as follows:

```
searchRx1.tuner1.tunerModel:    SI9136
searchRx1.tuner1.tunerInterfaceParm: 4096, 1, 0, 0, 5, 3, 0, 400
```

The second configuration file entries for tuner #2 are as follows:

```
searchRx1.tuner1.tunerModel:    SI9136
searchRx1.tuner1.tunerInterfaceParm: 4096, 2, 0, 0, 5, 3, 0, 400
```

```
   where:

   0 to 48896 in steps of 2048. Typical: 4096 VME Bus address
   1 or 2 = Channel
   0 (Independent), 1 (External), 2 (Slave) = LO Mode
   0 (Internal), 1 (External) = 10 MHZ Reference
   2 MHz to 20 MHz Default: 5 MHz = Start Frequency
   0 (Bypass), 1 (Enable) = Preselector Filter
   100 To 10000 uSec (400 Typical) = Settling
```

For directions on setting the address, see SI-9136B VHF/UHF Digital VME Tuner (page 50).

[1] VXI Mainframe shown for instructional purposes only.

Fiber Optic Cabling of the DSP Module

The proper routing of the FPDP fiber optic cables is dependant upon the type of receiver, ADC, and the configuration of the internal DSP modules. Some possible configurations of these internal modules are described in the section DSP Configuration (page 36). The following paragraphs describe and show cabling for several configurations:

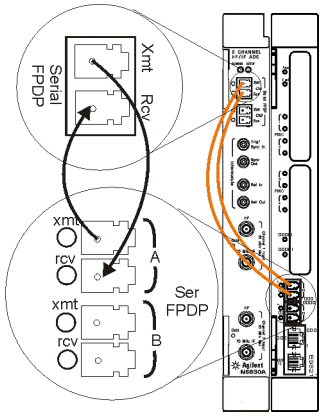
- N6830 HF or VHF/UHF Configuration, one DSP Page 29
- N6830 HF or VHF/UHF Dual Channel Configuration, one DSP per channel Page 30
- E1439D VHF/UHF Configuration, one DSP Page 31
- E1439D VHF/UHF Configuration, multiple DSPs with no delay Page 32
- E1439D VHF/UHF Configuration, multiple DSPs with delay Page 33
- E1437A HF Configuration, multiple DSPs with delay Page 34

N6830 HF or VHF/UHF Single Channel Configuration, One DSP

The following configuration shows one N6830 Dual channel HF Receiver and 70 MHz IF ADC and one E9821A DSP. This system is configured for search and for no delay. The cabling is the same whether the N6830 HF input or the N6830 70 MHz IF input is being used. As shown in figure 16, a pair of cables run from the N6830's Serial FPDP ports to the A section of serial ports on the E9821A DSP. The connections are made so that each XMT socket connects to the other card's Rcv socket.

The first FPDP cable runs from the N6830's Channel 1 XMT socket to the E9821A DSP's A section Rcv socket. The second FPDP cable runs from the E9821A's A section XMT socket to the N6830's Channel 1 Rcv socket.

Figure 16. FPDP Cables for N6830, Single Channel, One DSP, No Delay



For configuring multiple DSPs with no Delay, see "VHF/UHF Configuration, Multiple DSPs with no Delay" on page 32

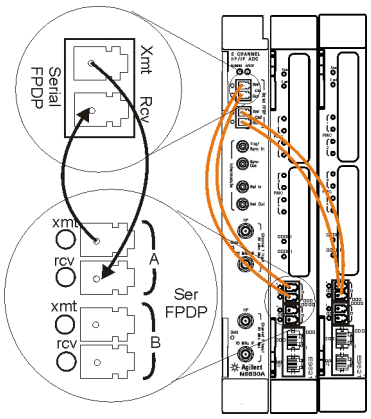
For configuring multiple DSPs with Delay, see "VHF/UHF Configuration, Multiple DSPs with Delay" on page 33

N6830 HF or VHF/UHF Dual Channel Configuration, One DSP per Channel

The following configuration shows one N6830 Dual channel HF Receiver and 70 MHz IF ADC and two E9821A DSPs. This system is configured for search and for no delay. The cabling is the same whether the N6830 HF inputs or the N6830 70 MHz IF inputs are being used. As shown in figure 17, a pair of cables run from the N6830's Serial FPDP ports to the A section of serial ports on both of the E9821A DSPs. The connections are made so that each XMT socket connects to the other card's Rcv socket.

The first FPDP cable runs from the N6830's Channel 1 XMT socket to the first E9821A DSP's A section Rcv socket. The second FPDP cable runs from the E9821A's A section XMT socket to the N6830's Channel 1 Rcv socket. The third FPDP cable runs from the N6830's Channel 2 XMT socket to the second E9821A DSP's A section Rcv socket. The fourth FPDP cable runs from the second E9821A's A section XMT socket to the N6830's Channel 2 Rcv socket.

Figure 17. FPDP Cables for Dual Channel N6830, Two DSPs, No Delay



Note

The two VHF/UHF search and collection systems are totally independent of each other. In this configuration, two instances of the E3238S application run in the same computer, so only one software license is required.

For configuring multiple DSPs with no Delay, see "VHF/UHF Configuration, Multiple DSPs with no Delay" on page 32

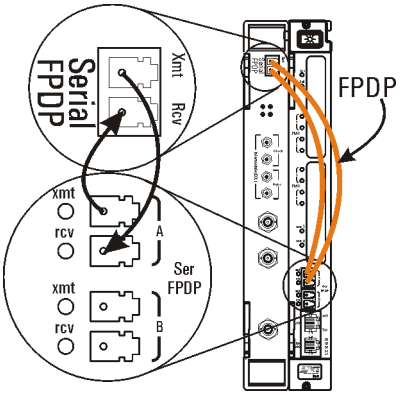
For configuring multiple DSPs with Delay, see "VHF/UHF Configuration, Multiple DSPs with Delay" on page 33

E1439D VHF/UHF Configuration, One DSP

The following configuration is that of one E1439D ADC and one E9821A DSP. This system is configured for search and for no delay. As shown in figure 18, a pair of cables run from the ADC's Serial FPDP ports to the A section of serial ports on the DSP. The connections are made so that each XMT socket connects to the other card's Rcv socket.

The first FPDP cable runs from the ADC's XMT socket to the DSP's A section Rcv socket; the second FPDP cable runs from the DSP's A section XMT socket to the ADC's Rcv socket.

Figure 18. FPDP Cables for VHF/UHF, One DSP, No Delay



For configuring multiple DSPs with no Delay, see "VHF/UHF Configuration, Multiple DSPs with no Delay" on page 32

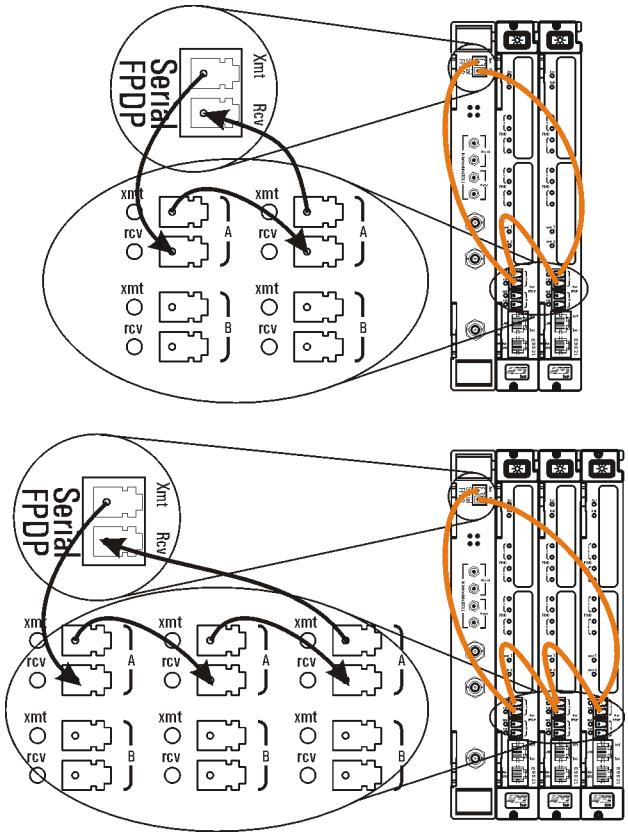
For configuring multiple DSPs with Delay, see "VHF/UHF Configuration, Multiple DSPs with Delay" on page 33

VHF/UHF Configuration, Multiple DSPs with no Delay

The following two configurations shown are that of a system with one E1439D ADC and two or three E9821A DSP modules, configured for search with no delay. See figure 19. This configuration daisy-chains the DSPs through their A section XMT and RCV sockets.

The upper part of figure 19 shows a two-DSP configuration. The ADCs XMT socket connects to the first DSP's A section Rcv socket. The first DSP's A section XMT socket is connected to the second DSP's A section Rcv socket. The second DSP's A section XMT socket loops back to the ADC's FPDP RCV socket.

Figure 19. FPDP Cables for VHF/UHF, Two or Three DSPs



The lower part of figure 19 shows a three-DSP configuration. The ADCs XMT socket connects to the first DSP's A section Rcv socket. The first DSP's A section XMT socket is connected to the second DSP's A section Rcv socket. The second DSP's A section XMT socket is connected to the third DSP's A section RCV socket. The third DSP's A section XMT socket finally loops back to the ADC's FPDP RCV socket.

For configuring multiple DSPs with Delay, see "VHF/UHF Configuration, Multiple DSPs with Delay" on page 33

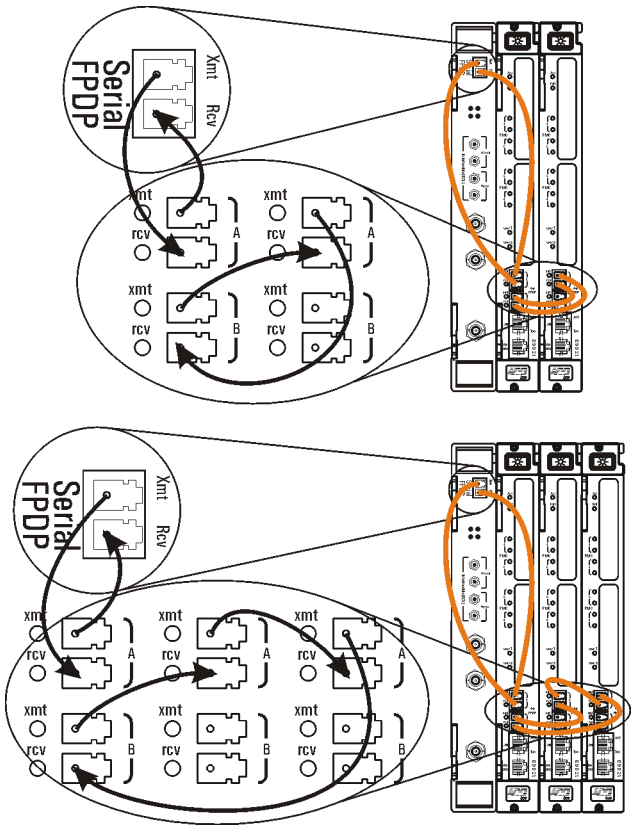
VHF/UHF Configuration, Multiple DSPs with Delay

The following subsection describe how to configure VHF/UHF systems for delay. In these configurations, the data is carried from the E1439D ADC to the DSP modules using fiber optic cables routed from the ADC's Serial FPDP port.

Two DSPs with Delay

The first configuration shown is that of a VHF/UHF system with one E1439D ADC and two E9821A DSP modules, configured for search and with delay. As shown in the upper configuration of figure 20, a pair of cables run from the ADC's Serial FPDP ports to the "A" set of serial ports on the first DSP. The connections are made so that each XMT socket connects to the other card's Rcv socket. Additionally, a second pair of cables connects the first DSP's "B" set of FPDP serial ports to the second DSP's "A" set of FPDP serial ports. Again, the connections are made so that each XMT socket connects to the other card's Rcv socket.

Figure 20. FPDP Cables for VHF/UHF, Multiple DSPs, Delay Configurations



Three DSPs with Delay

The next configuration has one E1439D ADC and three E9821A DSP modules, configured for search and a delay. As shown in lower configuration of figure 20, a pair of cables run from the ADC's Serial FPDP ports to the A section of serial ports on the first DSP. The connections are made so that each XMT socket connects to the other card's Rcv socket.

The remaining cables create a daisy-chain connection in the following manner. The first DSP's B section XMT connector connects to the second DSP's A section Rcv socket. The second DSP's A section XMT socket is connected to the third DSP's A section Rcv socket. Finally, the third DSP's A section XMT socket is connected back to the first DSP's B section RCV socket.

For configuring multiple DSPs with no Delay, see "VHF/UHF Configuration, Multiple DSPs with no Delay" on page 32

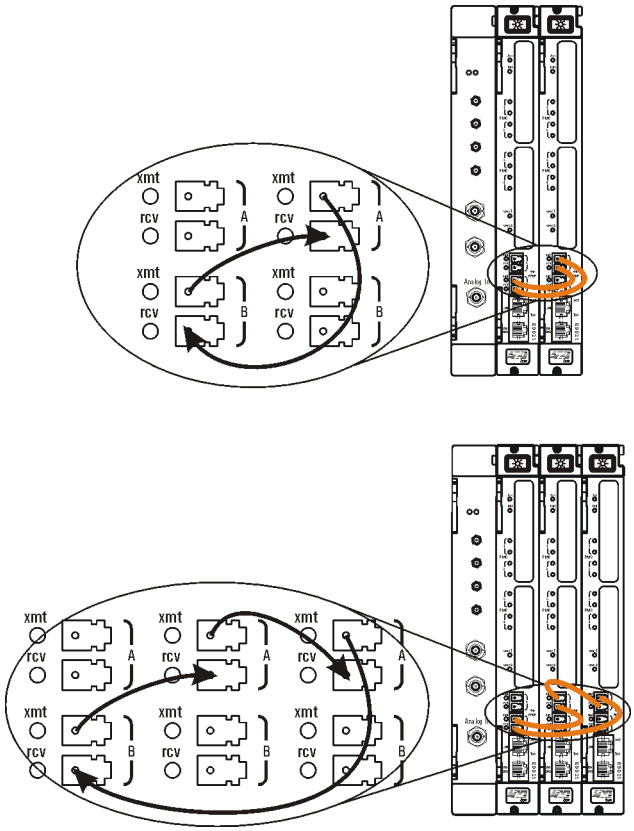
E1437A HF Configuration, Multiple DSPs with Delay

The following subsection describe how to configure E1437A HF systems for delay. In these configurations, the data is carried from the E1437A ADC to the first DSP module using the local bus. The data is then routed to and from later DSPs using fiber optic cabling.

Two DSPs with Delay

The first configuration shown is that of a HF system with two E9821A DSP modules, configured for search and with delay. As shown in the upper configuration of figure 20, a pair of cables connects the first DSP's "B" set of FPDP serial ports to the second DSP's "A" set of FPDP serial ports. Again, the connections are made so that each XMT socket connects to the other card's Rcv socket. The communications between the E1437A ADC and the first DSP occurs on the local bus.

Figure 21. FPDP Cables for HF, Two or Three DSPs, Delay Configuration



Three DSPs with Delay

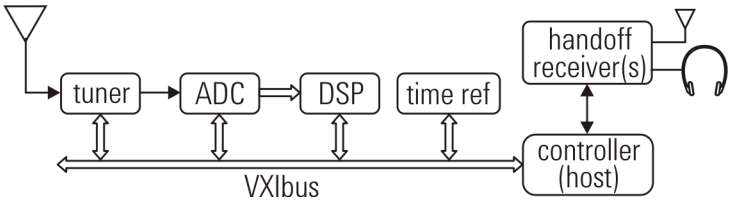
The next configuration has one E1437A ADC and three E9821A DSP modules, configured for search and a delay. The communications between the E1437A ADC and the first DSP occurs on the local bus. As shown in lower configuration of figure 20, a pair of cables create a daisy-chain connection in the following manner. The first DSP's B section XMT connector connects to the second DSP's A section Rcv socket. The second DSP's A section XMT socket is connected to the third DSP's A section Rcv socket. Finally, the third DSP's A section XMT socket is connected back to the first DSP's B section RCV socket.

Theory of Operation

The Block Diagram

This diagram illustrate how data flows in the system.

Figure 22. System Block Diagram



Tuner & ADC The tuner output is digitized by the ADC and passed to the DSP for search processing. The system sweeps the spectrum, sending blocks of magnitude data to the controller.

Note The tuner and ADC blocks are combined into one block when using the HF Receiver input of the N6830 Dual Channel HF Receiver and 70 MHz IF ADC.

Time Capture The ADC FIFO buffer may be used to collect time snapshot data. The size of the buffer depends on the ADC model and the options in it. For example the E1439D ADC has a 1 GB RAM option that serves the purpose.

Handoff Receivers As many as 100 handoff receivers can be managed by the E3238S system. These may be VXI modules (on the VXIbus) or connect via RS232, LAN, GPIB, etc.

Time Reference An IRIG card may be used to implement accurate time stamping or coordinate signal processing. See Page 44.

Digital Signal Processing This block may represent 1 or as many as 10 E9821A modules. The DSP configuration is discussed in more detail on Page 36.

DSP Configuration

Much of the detail of how the system works lies in the E9821A DSP module. It holds as many as 4 PMC modules[1]. The PMC modules are either G4 or digital down converters (DDC). A G4 module provides either **search (S)**, **signal processing (P)**, or **time delay (D)** functionality. The PMC module types are as follows:

- Dual-G4 (option 101) provides the maximum processing power for search or signal processing. It can also provide up to 11.5 seconds of delay in an HF system and 2.4 seconds in a VHF/UHF system.
- Digital downconverter (DDC, option 200) modules perform the *channelization* function. They provide as many as 32 narrowband channels each.

Each drawing shown here depicts a possible configuration for one E9821A module. The configurations are determined by two things:

- The types of modules installed on the E9821A provide the functional blocks.
- The library software determines how the data flows between the modules.

Figure 23. E9821 configuration examples for search and signal processing

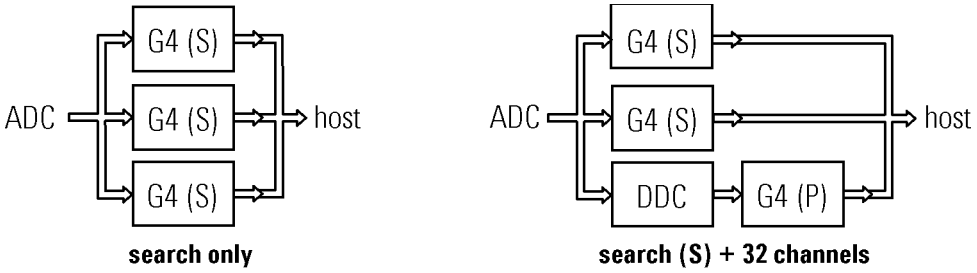


Figure 23 shows two configurations. On the left is a search (S) configuration optimized for sweep performance. The diagram on the right has both search and 32 channels of narrowband signal processing.

Figure 24. E9821 configuration examples for signal processing

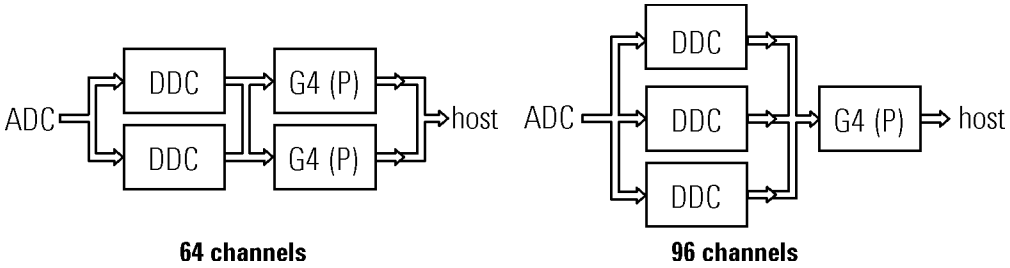
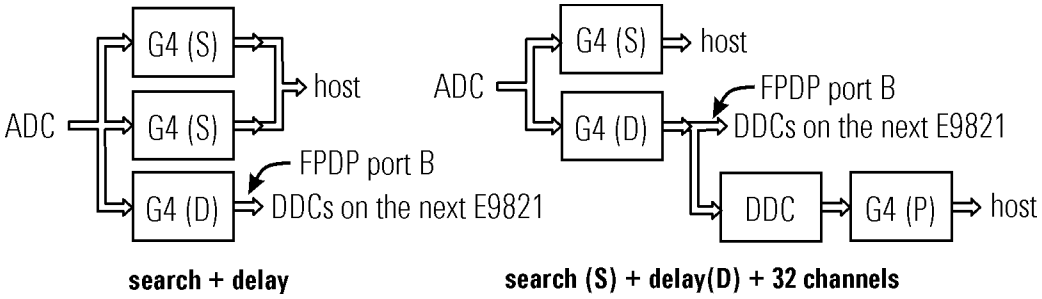


Figure 24 shows two E9821A signal processing configurations with various numbers of channels and processing power per channel. Time delay can also be implemented in a configuration shown in figure 25.

Figure 25. E9821A configuration for delayed signal processing

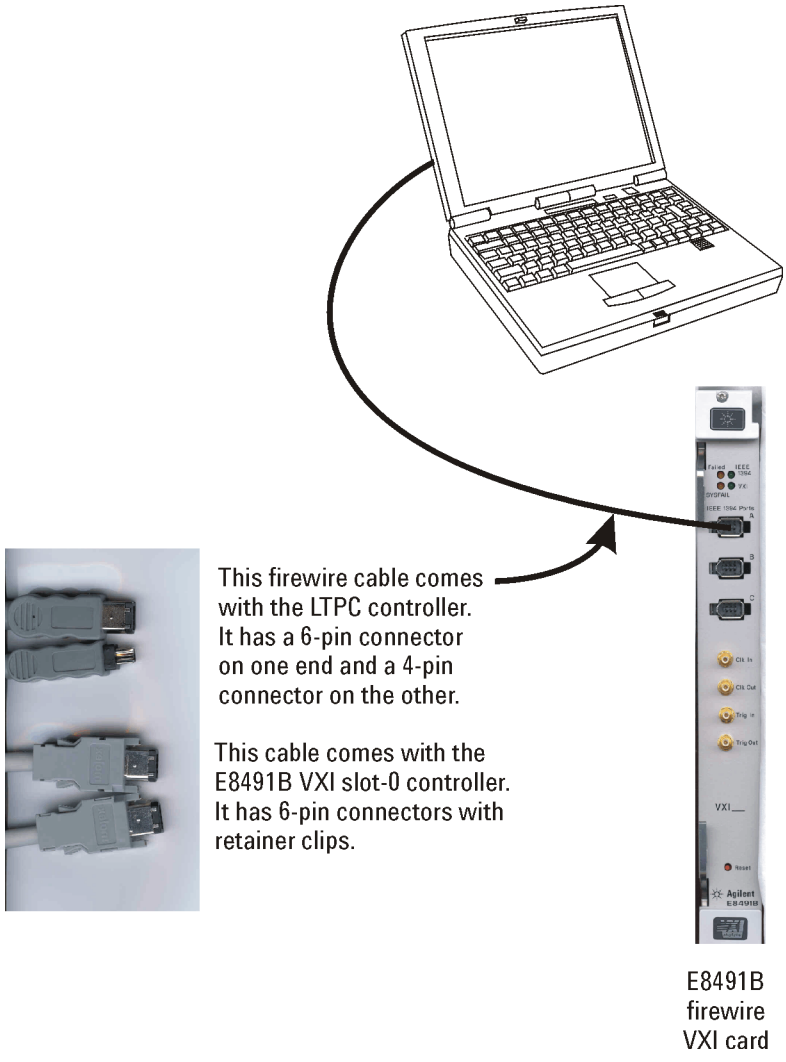


[1]Due to power and cooling limits, the maximum number of dual-G4 modules is 3.

Installing the System Controller

The laptop controller uses a firewire interface to connect to the VXI mainframe as shown in figure 26.

Figure 26. Laptop controller and firewire interface



Configuring VXI Components

This section covers configuration of the following VXI modules.

- ADC Configuration Page 39
- E9821A Signal Processing Page 42
- E1472A RF Multiplexer Page 43
- E1368/69A VXI Microwave Switch Page 44
- bc350 / 357 VXI-C Time and Frequency Processor (IRIG) Page 44

Tuner installation is described beginning on page 45.

Note If the system was integrated at the factory, this procedure can be skipped entirely.

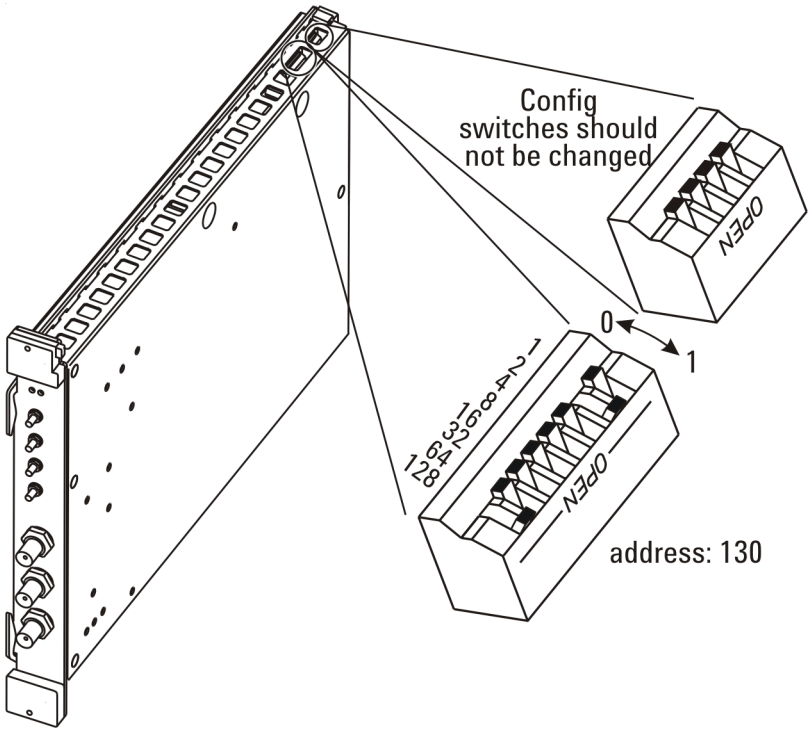
1. Before installing the VXI modules, turn off the power to the VXI mainframe and disconnect the power cord. This avoids damaging the modules during installation.
2. Set the DIP switches on the VXI modules as described on the following pages.
3. Check the connectors at the back of the VXI mainframe; they must have RFI boots installed (around the connectors) to meet performance specifications.
4. Firmly seat the modules in the VXI Mainframe with the ADC in the slot immediately to the left of the E9821A.
5. Secure the modules by tightening the captive screws that hold each module into the mainframe. This must be done to insure that performance specifications are met.
6. Edit the `e3238s.cfg` configuration file so that it describes the new configuration. The configuration commands used in it are described in the Hardware Configuration Reference on page 89.

Note The VXI logical addresses given in the module configuration instructions are recommendations. Any logical address may be used for any module as long as it is unique to that VXI chassis; i.e., each module must have a unique address.

E1437A, E1438A/B/D, E1439A/B/D, N6830 VXI ADC

ADC modules that use the VXI local bus to transfer data to the DSP module must be installed in the VXI slot immediately left of the DSP module. ADC modules that use FPDP do not have this constraint.

Figure 27. ADC Logical Address Setting



Note

The N6830 only has an address switch. It does not have a configuration switch. The address switch sets the N6830 channel one address. The channel two address is the channel one address plus 1. In the figure above, the channel one address would be 130. The channel two address would be 131.

Note

The N6830 has a front panel error LED which will blink in the following circumstances:
1. The selected VXI logical address is not valid (0, 254, 255 are invalid). The LED will be lit continuously.
2. An internal error, such as an unlocked PLL, has occurred. The LED will blink 3 times per second.
3. An external reference is programmed, but the reference PLL is not locked, or no external signal is present, or the external reference is low amplitude or has a non-symmetrical duty cycle. In this case, the LED will blink 1.5 times per second.
4. A framing error from one of the ADC deserializers was detected. In this case, the LED will blink twice and then pause.

The following information is taken from the d.e3238s.cfg file. Use this information to help configure the ADC settings in the e3238s.cfg file for your system.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!              Search Receiver ADC Configuration                !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

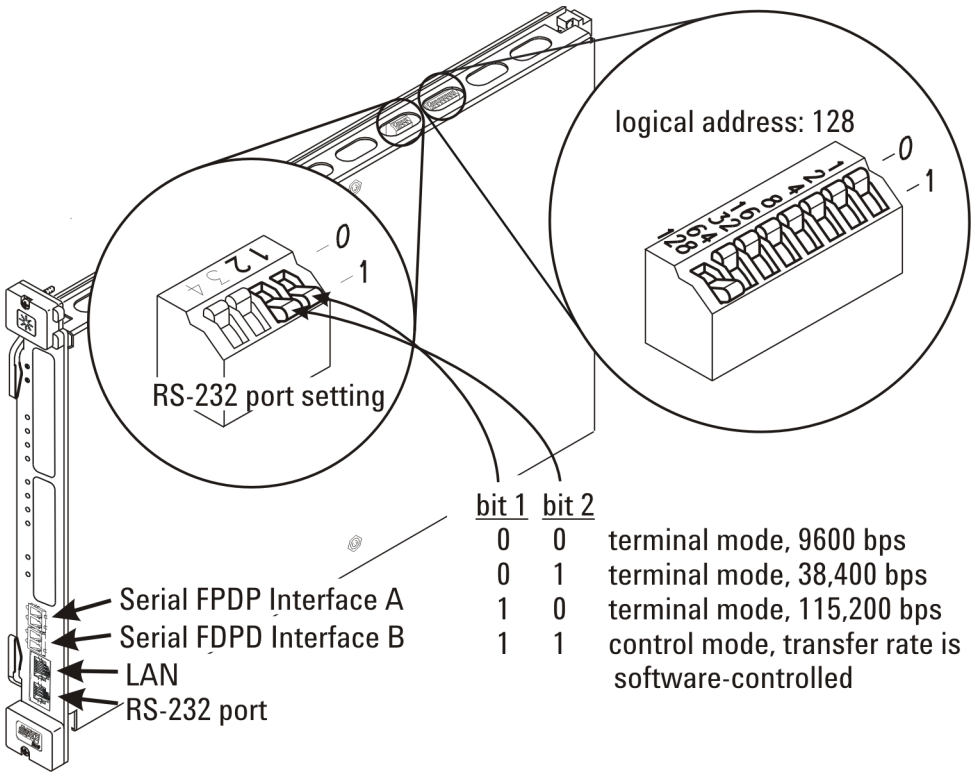
```
! The ADC configuration commands are as follows:
!  - adcModel [E1437A, E1438A, E1438B, E1439A/70, E1439A/BB, E1439B/70,
!              E1439B/BB, E1439C/70, E1439C/BB, E1439D/70, E1439D/BB,
!              N6830A/70, N6830A/HF]
!       Specifies the ADC's model type.
!       If the ADC model type is N6830A/HF, only the HF input of the N6830
!       module is used and the tunerType must be set to N6830A/HF.
!  - adcInterfaceParm  [1 to 254]
!       Specifies the ADC's logical address.
!
!  - adcClock  [Internal, External]
!       specifies whether to use an internal or external ADC clock.
!       The clock must be 20.48 MHz for the E1437A module.  For the E1438,
!       E1439, or N6830 modules, the frequency must be 10 MHz, and it will
!       be used as a reference to derive the actual sample clock.  The clock
!       must be present at the clock input (or Ref Input) before the software
!       is started.
!  - adcDataPort  [localBus, FPDP, VXI]
!       Specifies the data path between the ADC and DSP module. The N6830A
!       module only supports FPDP.
!  - adcMasterClock [Off, On, Auto]
!       Specifies whether the ADC puts its clock signal on the VXI backplane
!       for use by other modules.  Only one ADC may drive the VXI backplane.
!       If the Auto mode is selected, the adcMasterClock for the search ADC
!       is turned Off unless the searchRxConfiguration is set to a multiple
!       channel mode. In this case the first ADC is set to On and the other
!       ADC's are set to off.
!  - adcSampleRate [10240000, 20480000, 40960000, 81920000, etc (see below)]
!       Specifies the ADC sample rate for the N6830A.  This parameter will
!       affect the bandwidth available for narrowband signal processing.
!       For best probability of intercept (fastest search revisit times) use
!       the lowest stare bandwidth that covers the frequency range of
!       interest.
!         N6830A/HF
!         Sample Rate     Stare Bandwidth
!         81920000        32 MHz
!         40960000        16 MHz
!         20480000         8 MHz
!         10240000         4 MHz
!         N6830A/70
!         Sample Rate     Stare Bandwidth
!         95000000        36 MHz
!         47500000        18 MHz
!         23750000         9 MHz
!         11875000       4.5 MHz
!       If you are upgrading from a 9119-1 tuner with E1437 ADC
!       to an N6830A/HF, use the following settings.
```

```
!          searchRx1.adcModel:      N6830A/HF
!          searchRx1.adcSampleRate: 20480000
!          searchRx1.adcDataPort:   FPDP

!        Increasing the ADC sample rate may affect the sweep rate depending
!        on the search setup number of averages and RBW selections.

searchRx1.adcModel:           E1439D/70
searchRx1.adcInterfaceParm:   130
searchRx1.adcClock:           Internal
searchRx1.adcDataPort:        FPDP
searchRx1.adcMasterClock:     Auto
```

E9821A VXI Signal Processing Module

There may be as many as ten E9821A modules in a system. The capabilities of the module depend on the type and number of PMC cards installed on it. The first module is configured to provide search processing but may also provide some signal processing capability. Search processing requires G4 modules (only). Signal processing requires both DDC and G4 modules. See page 36.

Figure 28. E9821 DSP



RS-232

The RS-232 port may be used to control an external tuner such as the 89431. A special cable comes with the 89431 for use with this module. See page 45.

The RS-232 port should be set to provide command control (bits 1 and 2 set to 1)

LAN

The LAN port is used for development debugging, service trouble shooting, and to control the PSA tuner. See page 24.

Power and cooling limits

The E9821A is a carrier for PMC and ePMC plug-in boards. There are four sites for ePMC boards. Due to power and cooling limitations, **no more than three[1] dual-G4 processing boards may be installed in a module**.

All unoccupied VXI chassis slots should have front-panel covers (part number E8400-60202) installed when the system is on. Also, chassis with selectable fan speed should have it set to "High."

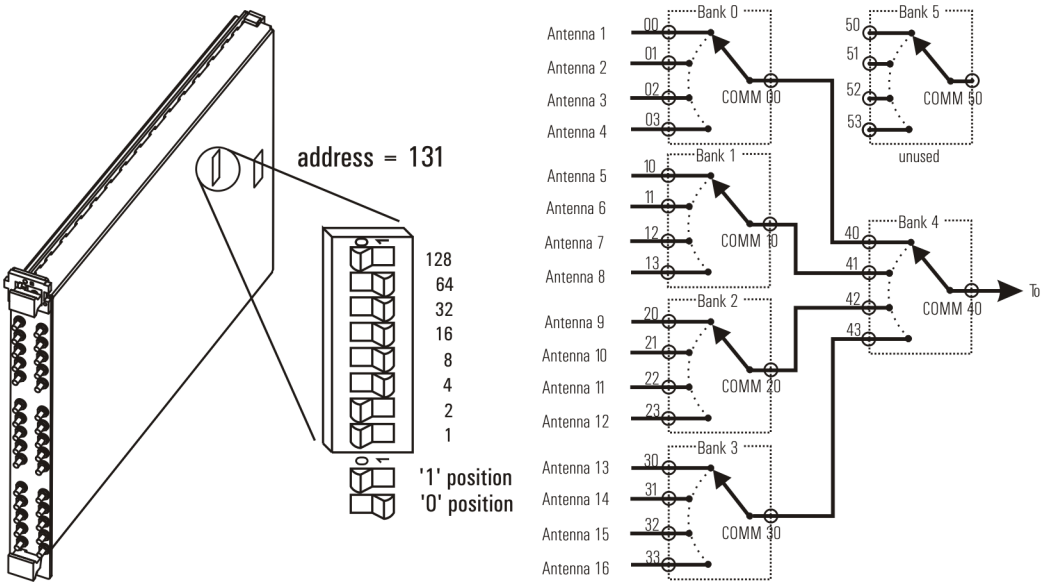
When the E9821A is used in the 4-slot chassis (E8408A), no more than *one* dual-G4 board should be installed.

[1]There should be no more than two (2) dual-G4 cards in E9821A modules that have serial numbers less than US43140000.

E1472A VXI RF Multiplexer

The E1472A VXI RF Multiplexer module may be installed in any slot.
The E1472A logical address should be set to 131.

Figure 29.
E1427A Settings



The E1474A is identical to the E1472A except that it has 75Ω connections.

Note

Only one switch module may installed in a system.

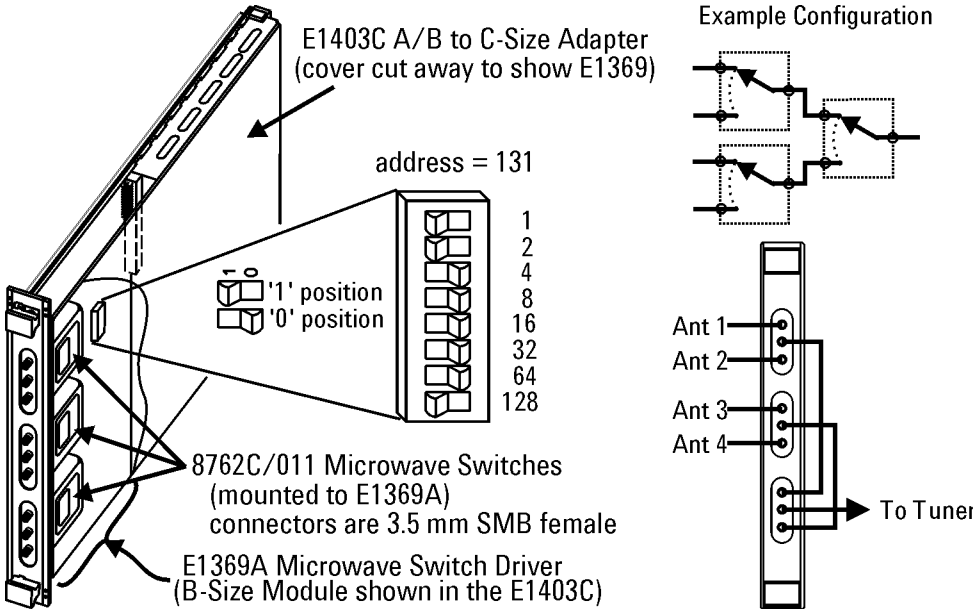
The `e3238s.cfg` (default) configuration file contains the commands to implement the switch configuration shown above. To enable this configuration, just remove the comment characters in the first column of these commands.

This module may also be used to implement multiple tuners; see searchRx.tuner.tunerSwitchCmnd (pg 158).

E1368/69A VXI Microwave Switch

This VXI module supports switching signals for the microwave tuners. It can be installed in any chassis slot. Set the address to 131.

Figure 30. E1369A Settings

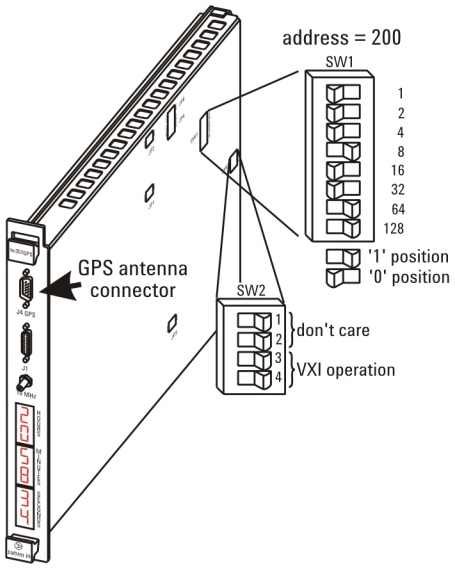


The configuration shown above consists of the switch driver board, the switches, and an extender module. It may be ordered as special option 404-E3238B. Cabling for four antennas into one tuner is shown above.

bc350/357VXI-C Time and Frequency Processor (IRIG)

This VXI module provides accurate time stamp information. The model bc357VXI also has a GPS satellite receiver for a source. This module may be installed in any chassis slot. Set the address to 200.

Figure 31. bc350VXI Settings



Jumpers	
JP1 time code	1-2: DC levelshift
	3-4: modulated
JP2 GPS mode	1-2: single-ended 1pps
	3-4: differential 1pps
JP3 GPS sensor	1-2: ACUTIME Smart Antenna or SV-6
	3-4: TANS
JP4 RS422 mode	3-4: 5-6: have an aux. RS422 output port
	3-4: RS422 ports are daisy-chained 5-6: such that IRIG signal is passed on
JP5 RS422 load	1-2: puts 100W load between diff input lines
	3-4: no load added

Tuner Configuration

This section cover the installation of the following components:

- 89431A tuner (2.65 GHz)
- Watkins-Johnson WJ9119 VXI tuner (0.5–32 MHz) Page 47
- E2730/31B VXI tuner (20 MHz–2.7 GHz/6 GHz) Page 48
- DRS SI9250 VXI block down converter Page 49
- DRS SI9136B VXI UHF/VHF Digital VME Tuner Page 50
- Communications Solutions CS-5040 VXI tuner (18 GHz, 40 GHz, 60 GHz) Page 52

89431A Tuner

Note

If your system does not include a 89431A tuner, this procedure is not applicable.

The installation of the 89431A RF tuner requires the AFU cable kit. This kit contains the RS-232 cable that connects the tuner to the E9821A and a 50Ω BNC IF signal cable that connects the tuner to the ADC. (See figure 33.)

1. Turn power off.

   Turn the power to the RF tuner off and disconnect the power cord before installing or configuring the 89431A to avoid damage.

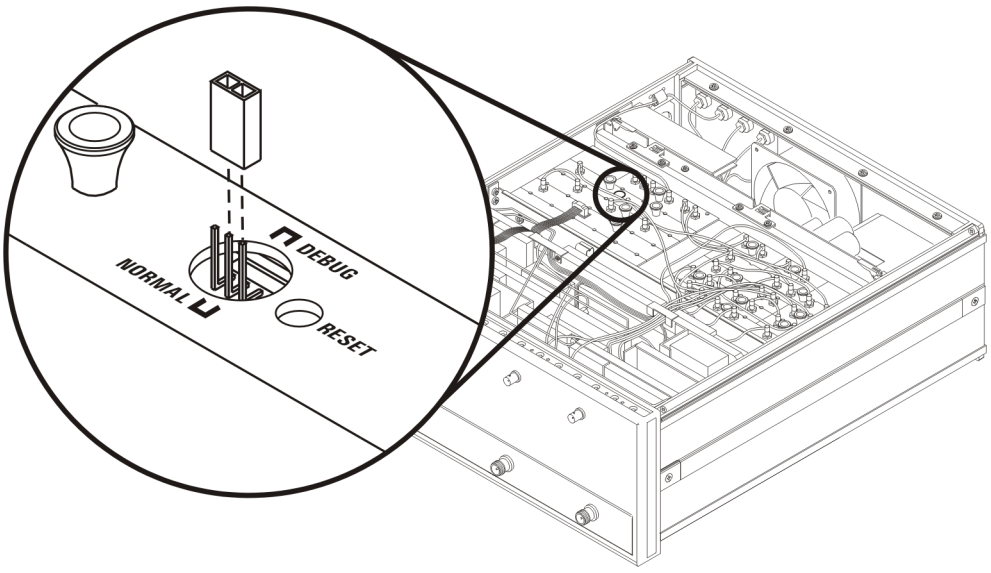
2. Set the baud rate jumper.

   If the system was integrated at the factory, skip this step and proceed to step 3.

   Before the 89431A RF tuner can be used in the E3238S system, an internal jumper must be set to the proper position.

   To access this jumper, remove the top cover from the tuner and set the jumper in the NORMAL position as shown in figure 32. This corresponds to a baud rate of 125000 which must be specified in the `e3238s.cfg` file.

Figure 32. Move jumper to the NORMAL position



When the jumper is in the correct position, replace the top cover and securely tighten the retaining screw.

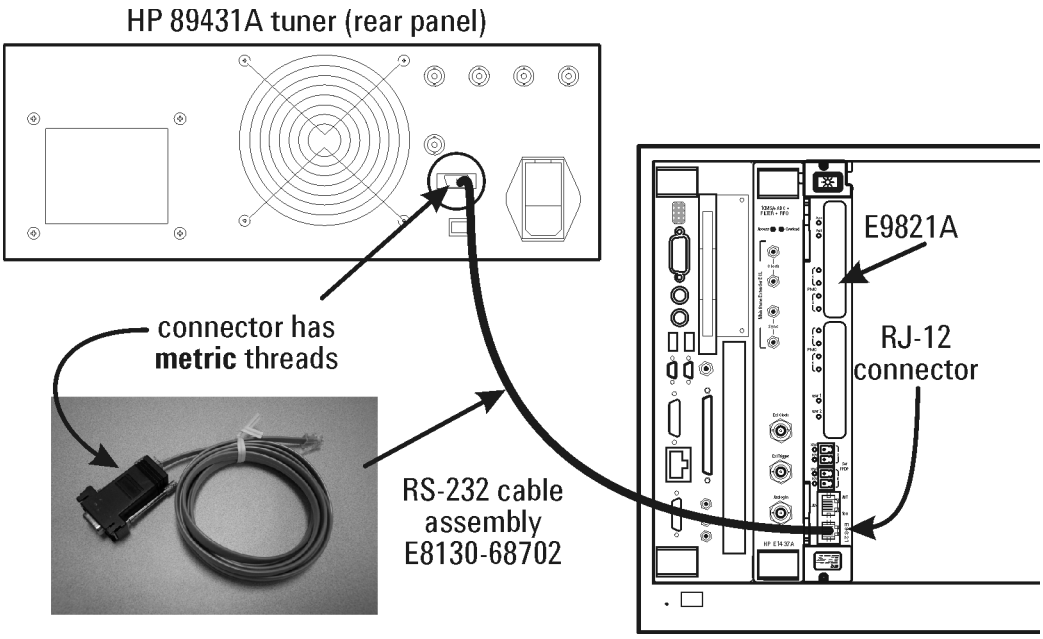
3. Connect the RS-232 cable.

   The 89431 tuner is connected to theE9821A module by an RS-232 cable included in the AFU kit. See figure 33.

Caution

The RS-232 connector on the 89431 has metric threads. Be sure to use the proper adapter to avoid damaging the threads.

Figure 33. Connecting the RS-232 cable



   Attach the cable as shown above.

4. Connect the IF signal cable.

   The 89431A RF tuner is connected to the ADC module by a 50Ω BNC cable included in the AFU kit.

   a. Attach one end of the BNC cable to the front panel connection on the 89431A RF tuner labeled “OUT (to channel 1)”.

   b. Attach the other end of the BNC cable to the front panel connection on the ADC module labeled “Analog In”.

5. Reconnect the power cable.

This concludes the hardware installation procedure for the E3238S system.

See searchRx.tuner.tunerInterfaceParm (pg 156) for the configuration command that specifies the tuner model.

Watkins-Johnson WJ9119-1 VXI HF Tuners

Note

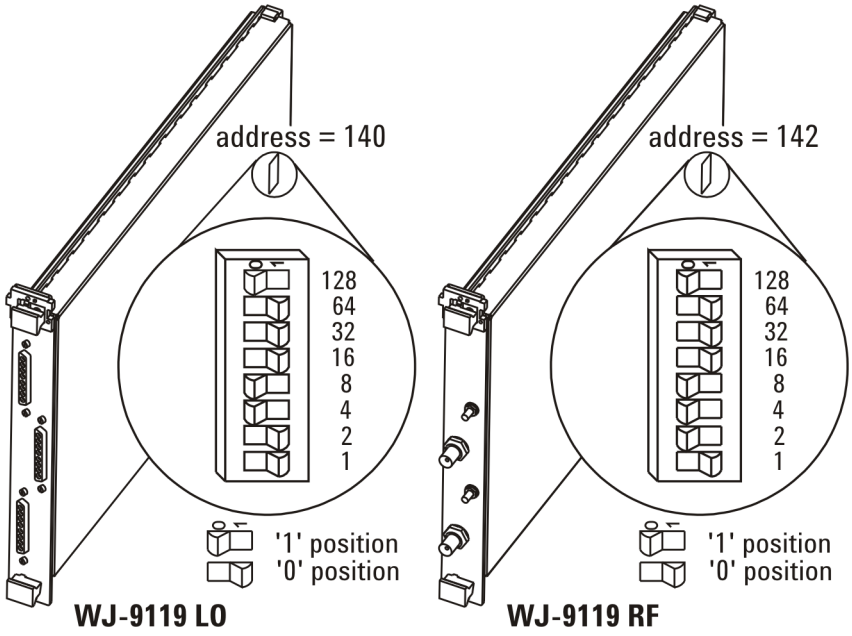
If your system does not include a WJ9119-1 tuner, this procedure is not applicable to your installation.

Configure the modules as described in figure 34 and then install them in the VXI mainframe as described in figure 9.

- Set the logical address of the LO module to 140.
- Set the logical address of the RF module to 142.

See searchRx.tuner.tunerInterfaceParm (pg 156) for the configuration command that specifies the tuner model.

Figure 34. Configuring the WJ9119-1 tuner



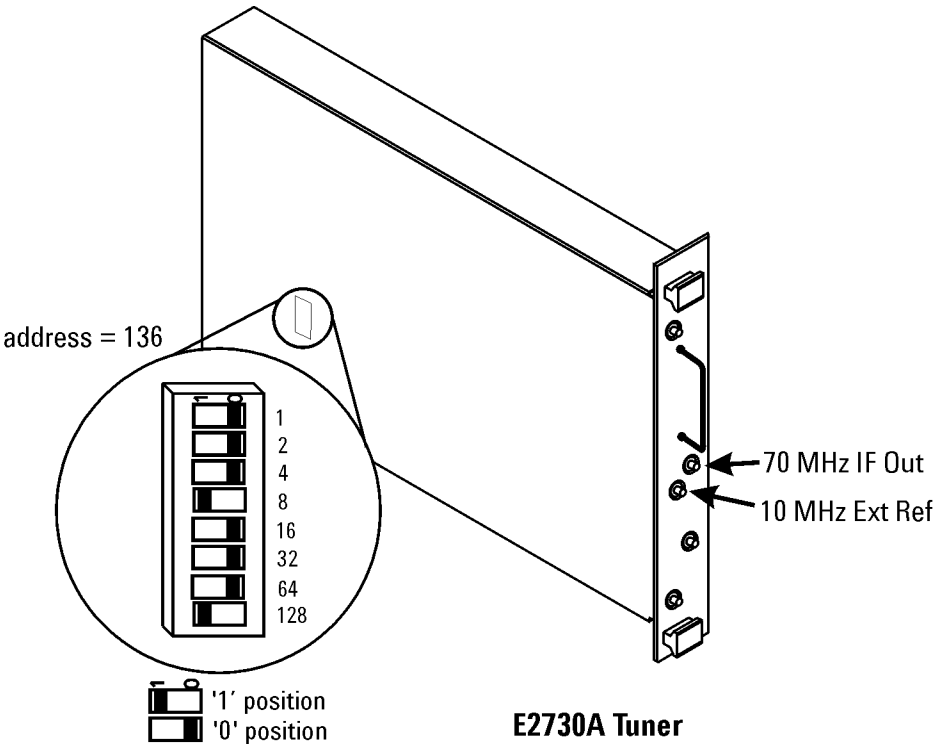
E2730B/E2731B VHF/UHF Tuner

Note If your system does not include an E2730B or an E2731B tuner, this procedure is not applicable to your installation.

Set the module's logical address as described in figure 35 and then install it in the VXI mainframe as described in figure 9 on page 22.

See searchRx.tuner.tunerInterfaceParm (pg 156) for the configuration command that specifies the tuner model.

Figure 35. Configuring the E2730B tuner



Note When a 10 MHz reference signal is connected to the external reference input, the configuration file entry for this module must be set accordingly. Failure to do so may cause frequency accuracy problems.

See searchRx.tuner.tunerInterfaceParm (page 156)

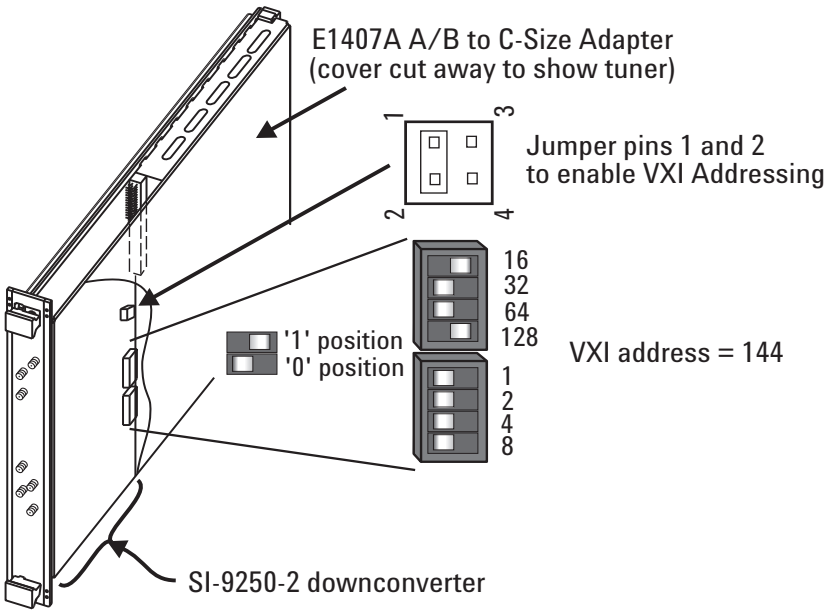
SI-9250 Block Downconverter

Note

If your system does not include an SI-9250 tuner, this procedure does not apply to your installation.

Set the module's logical address as described in figure 36 and then install it in the VXI mainframe as described in figure 13 on page 25.

Figure 36. Configuring the SI-9250 tuner



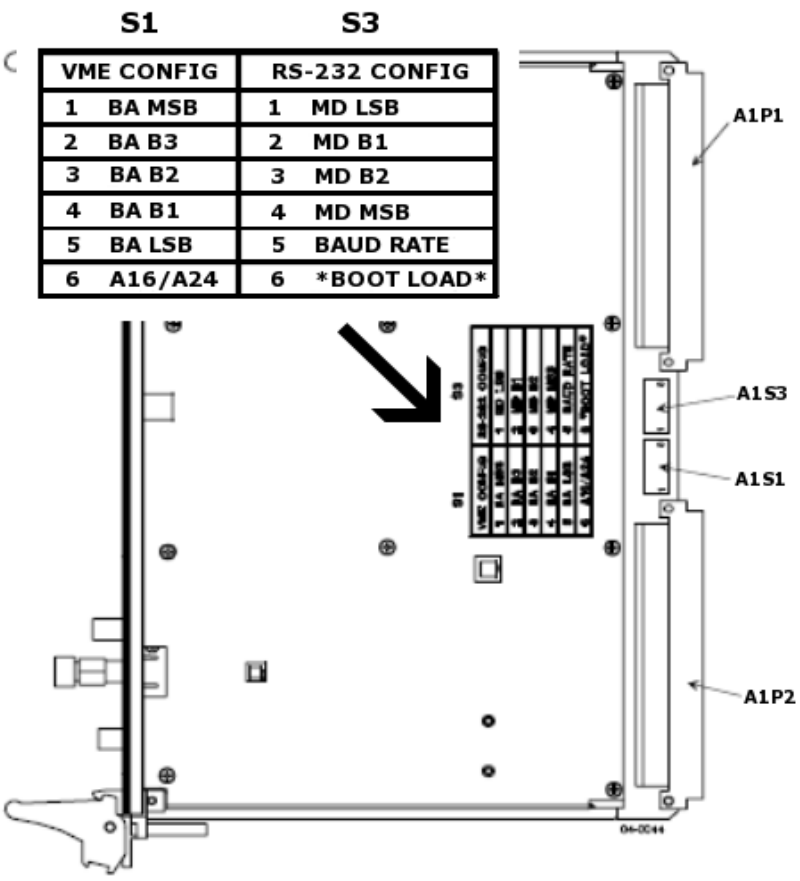
The configuration file entry for this module is given on page 25.

SI-9136B VHF/UHF Digital VME Tuner

If your system does not include an SI-9136B tuner, this procedure does not apply to your installation.

Set the module's VME address as described in figure 37 and then install it in the VXI mainframe using the required VME/VXI carrier as described in figure 14 on page 26.

Figure 37. Configuring the SI-9136B tuner



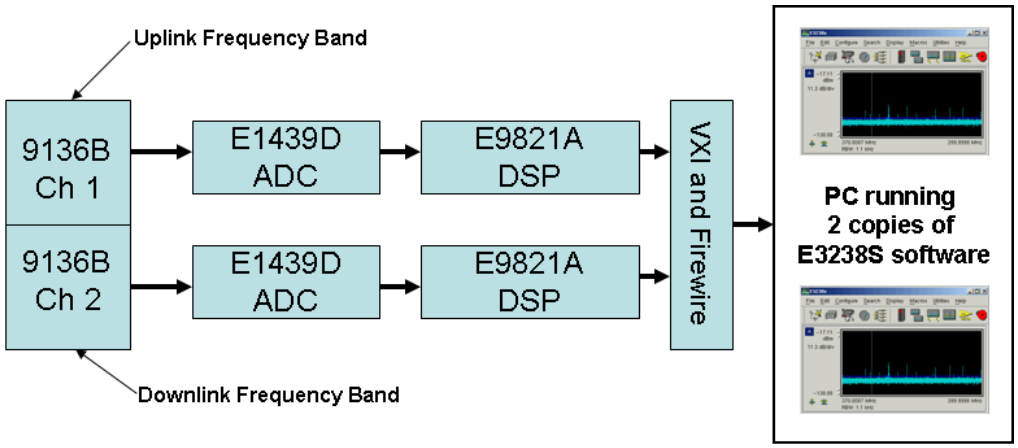
The configuration file entry for this module is given on page 26.

Note

For more details on VME Config and RS-232 switch settings, please consult the SI-9136B Manual.

Version E2.20 and newer of the E3238S software supports independent use of the SI-9136B dual tuner channels. The diagram in figure 38 illustrates how the dual tuner can be used by two instances of the E3238S software in a Dual Band configuration. This configuration allows collection in two different RF frequency bands.

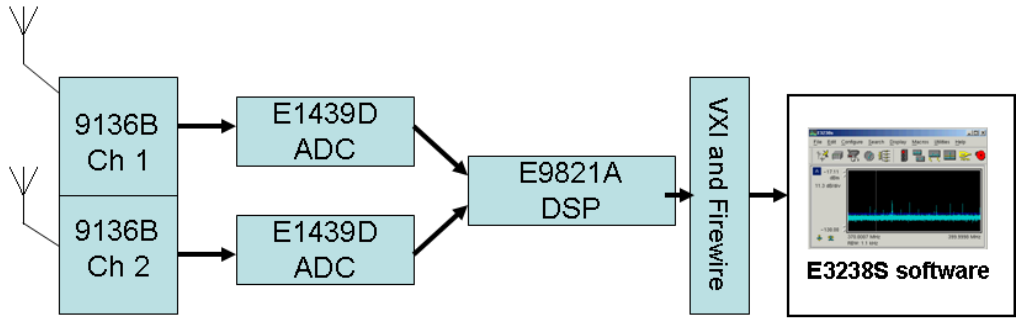
Figure 38. Dual Tuners with 2 E3238S Instances



Dual Band Configuration

The diagram in figure 39 illustrates the SI-9136B in a 2 search receiver configuration. In this configuration, both receivers are tuned to the same frequency.

Figure 39. Dual Tuners in a 2 Search Receiver Configuration

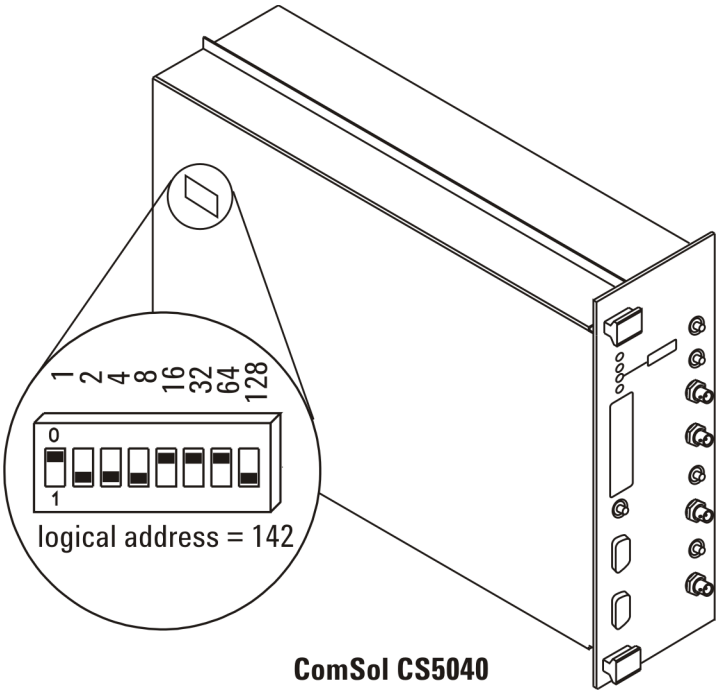


2 Search Receiver Configuration

Communication Solutions CS-5040 Microwave Tuner

This tuner is used with the E1439A/B/D ADC as shown in figure 9 on page 22.

Figure 40. Configuring the CS-5040 tuner.



Wideband Recording and Playback Sub-System Configuration

This section covers the installation of the Conduant Big River Model LTX2 (E3238S-050) or LTX2-35 (E3238S-051) wideband recorders.

The Wideband Recording and Playback Sub-System (WRP) connects to any E3238S system via Ethernet for control, and via optical S-FPDP (VITA 17.1 optical) for data. The Ethernet port connects via a LAN crossover cable to the E9821A DSP, which issues the commands (Record, Playback, etc.) for the WRP. The recorder is inserted into the data path between the ADC module (Agilent E1439D ADC or Agilent N6830A dual channel HF receiver and 70 MHz IF digitizer) and a DSP module (Agilent E9821A DSP).

Configuration File Modification

Update the Agilent E3238S Signal Detection and Monitoring Solutions *configuration file* for Agilent 35688E Option WRP, Wideband Recording and Playback Subsystem.

The following text assumes that the Agilent E3238S Signal Detection and Monitoring Solutions is installed in the default directory `C:\E3238S`.

To use the Agilent 35688E Option WRP, Wideband Recording and Playback Subsystem, the LAN interface must be enabled in the DSP module so that the DSP module auto recognizes the disk hardware.

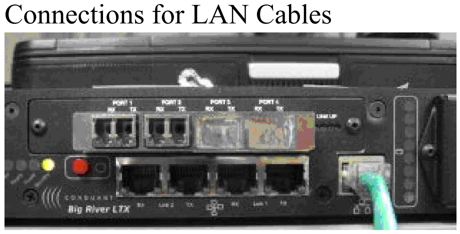
The following lines appear in the E3238s *configuration file.* Use the IP address of your WRP for the searchRx1.wrpIpAddress.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!         Wideband Record/Playback Configuration          !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! !!!
!    wrpIpAddress:          LAN IP address of the Conduant Data
!                             recorder in dot notation.!
searchRx1.wrpIpAddress:      10.1.249.101
```

LAN Connection

The recorder is connected via LAN to the Agilent E9821 DSP module. If the recorder is the only LAN device that must be connected to the Agilent E9821 DSP, the connection cable must be a LAN "crossover" cable. This connection is shown by the green cable in the photo below.

Connections for LAN Cables



If the configuration requires multiple LAN devices to be connected to the Agilent E9821 DSP LAN port, a 100-baseT LAN hub or switch must be used. In this case, crossover cables cannot be used. Configurations that use an Agilent PSA as a tuner, but also require an Agilent 35688E Option WRP, Wideband Recording and Playback Subsystem would encounter this configuration.

Serial FPDP Cable Connections

The serial FPDP connector labeled "Port 1" on the recorder is connected to the ADC (Agilent E1439 or Agilent N6830)

Connections for Serial FPDP Cables

This is a full duplex connection -- Rx to Tx and Tx to Rx.

The "Port 2" connector is connected to the Agilent E9821 DSP "Port A" serial FPDP connector. If the configuration is using a single Agilent E9821 DSP, a simple duplex connection is made -- Rx to Tx and Tx to Rx.

If the configuration has multiple Agilent E9821 DSPs, the cabling depends on whether delay is being used for narrowband processing (specified in the *configuration file*, for example, `searchRx1:minDelayTimeRequired:0.5`).

- If delay **is not** being used, the FPDP cables are connected as shown in the *Agilent E3238S Installation and Configuration Reference* in the VHF/UHF Single Channel Configuration, Multiple DSPs with **no** Delay section, except the ADC connection is connected to "Port 2" of the recorder. In this configuration, sFPDP “Port B” on the Agilent E9821 DSP modules are not used.
- If delay **is** being used, the FPDP cables are connected as shown in the *Agilent E3238S Installation and Configuration Reference* in the VHF/UHF Single Channel Configuration, Multiple DSPs with Delay section, except the ADC connection is connected to "Port 2" of the recorder.

LAN Configuration -- Recorder/Agilent E9821 DSP Devices Only

The network IP addresses for the recorder and the Agilent E9821 DSP have been set when configured at the factory for new systems. The recorder's address is 10.1.249.101 and the Agilent E9821 DSP's address is 10.1.249.100. Connect the two LAN ports using a crossover cable (also referred to as a direct connection or loopback cable). You can also use a standard LAN cable with a crossover adaptor.

If you're adding recording capability to an existing E3238s system, the LAN port on the Agilent E9821 DSP must have its IP address set and the LAN port enabled. The `e9821LanConfigure.exe` utility program in the `/e3238s/bin` folder is used for this configuration. Using this utility, set the IP address to 10.1.249.100 and set the netmask to 255.255.0.0.

LAN Configuration -- Recorder, Agilent E9821 DSP, Other Device (for example, PSA Spectrum Analyzer)

If the configuration includes another LAN device, such as an Agilent E444x PSA being used as a tuner, a networking hub or switch must be used. The hub or switch will create a private network that is typically isolated from all other LANs. For this configuration, set the IP address of the PSA to 10.1.249.102. When a network hub or switch is used, standard LAN cables are used (crossover cables will not work).

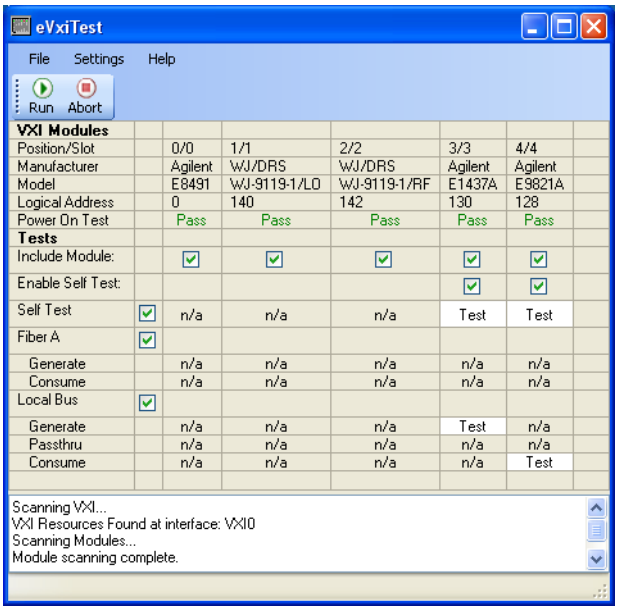
Hardware Diagnostics

Testing VXI Modules

A hardware diagnostic utility called eVxiTest is provided. This utility is installed when you install the E3238S software. To run the utility, click **Start > All Programs > Agilent E3238S > eVxiTest > eVxiTest**.

When you run the program, a window appears as shown in figure 41.

Figure 41. Agilent eVxiTest application



This utility identifies and performs a power on test on all VXI modules found in the VXI mainframe.

Click **Run** to perform a self test on all modules with self test capability. Units with self tests have the word *Test* in their Self Test field. Click **Help > Overview** for more eVxiTest information.

Note

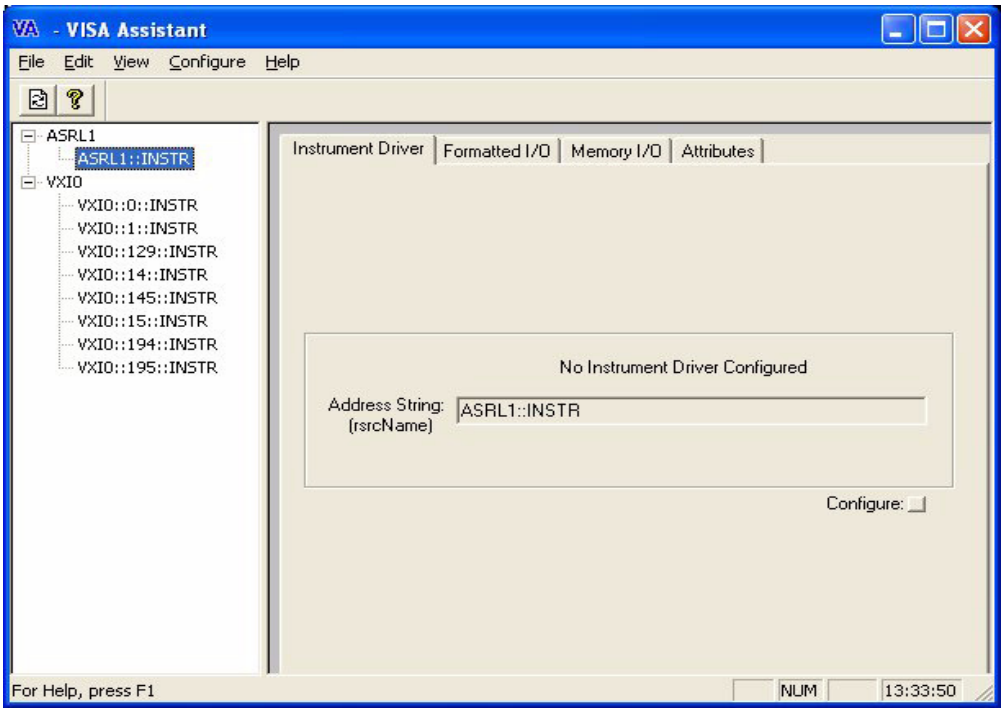
Halt any program that accesses the VXI modules before running this utility. *This includes the E3238S application.*

Laptop/Firewire System Diagnostics

The VISA Assistant displays the system configuration and provides information about the various VXI modules installed in the mainframe such as the logical address, and attributes. See Configuring the VXI interface (page 70).

To start it click **Start > Programs > Agilent IO Libraries Suite > Utilities > Visa assistant**

Figure 42. Agilent Visa Assistant



Compare the addresses here with the entries in the `e3238s.cfg` file. Also, check the E3238S **File > Shared Libraries** dialog box to determine which `e3238s.cfg` file is used.

Software Installation and Configuration

Software installation

This chapter describes the installation and configuration of the software for the E3238S Signal Detection and Monitoring System.

For new systems - Factory Integrated

If your system was integrated at the factory, your system is ready to operate. The only installation required is to connect cables and insert the license key in a free USB port if your software license uses a USB key. We recommend that you immediately make a backup copy of the software license file.

For new systems - User Integrated

To install the E3238S software on a commodity laptop, follow the installation procedure described in Installing and Configuring the E3238S Software (page 65). This procedure should handle most installation issues automatically. The minimum and recommended controller requirements are listed in Controller Requirements (page 61)

For Upgrades

To upgrade the E3238S software from a previous version just run the `setup.exe` program; see Installing and Configuring the E3238S Software (page 65). This procedure should handle most installation issues automatically.

N6841A RF Sensor

For software and hardware installation information, refer to the manuals that came with the N6841A RF Sensor.

This chapter contains the following topics:

Updating System Components

Various files used in the previous version may need to be changed before the new version can be used.

- Configuration files (such as e3238s.cfg)
- Resource file (the E3238s file, if used)
- Custom library extensions

Configuration File Update

When the installation program `setup.exe` is run, a new configuration file is installed as \E3238s\d.e3238s.cfg (overwriting the previous copy). You should compare the configuration file(s) in use against this one and convert the usage as appropriate to be compatible with new capabilities.

Resource File Update

The application resource file \E3238s\d.E3238s should also be compared to the resource file in use (E3238s).

Updating Custom Libraries

It may be necessary to recompile any custom libraries which you have created. An upgrade note describing the details should accompany the material sent as an upgrade kit.

New versions of libraries purchased from Agilent should be delivered to you on separate CDs as part of the upgrade.

For System Recovery

To recover the system from something like replacement of the system disk, first recover the controller operating system using the recovery disks provided with the controller and then perform an installation with the most recent version of the E3238S software. See Installing and Configuring the E3238S Software (page 65).

Note

The E3238S software requires a valid copy of your software license file. If you do not have a valid license file, contact Agilent to have one regenerated and e-mailed to you. See The Software License File (page 76).

Controller Requirements

The following table lists the minimum and recommended requirements for a system controller.

CPU	MINIMUM	RECOMMENDED
Processor	1.5 GHz Pentium P4 (E3238S libraries are optimized for Intel Processors)	Dual 3 GHz Pentium (*Load eXceed on one CPU & E3238S on the other)*
Memory	512 MB (performance suffers with less)	2 GB (More memory is always better)
Operating System:	Win2k w/SP4 or XP w/SP2 These are the only operating systems that have been E3238 tested	Win2k w/SP4 or XP w/SP2 These are the only operating systems that have been E3238 tested
Drives	**MINIMUM**	**RECOMMENDED**
Hard Drive	20 GB (400MB required for installation of E3238S SW)	120 GB (400MB required for application software) (SCSI or Raid0 faster for snapshots)
Floppy Disk Drive:	Not required by E3238S system (May be required for system recovery & boot floppies)	1.44MB Floppy Drive (May be required for system recovery & boot floppies)
DVD Drive	DVD drive (Needed to install E3238S software and License file)	16x DVD+RW (Used to install and backup software
Graphics	**MINIMUM**	**RECOMMENDED**
Display	17" Display Required to view spectral data points	20" LCD Display Required to view spectral data points
Graphic Card	True 1024x1280 (on-screen resolution) 16 Bit True Color 8 Mbyte On-board Video Memory (Required for E3238S high speed color displays)	True 1600x1600 (on-screen resolution) 32 bit True Color AGP Video Card with 128 Mbyte memory (Required for E3238S high speed color displays)

Communication (I/O)	MINIMUM	RECOMMENDED
Sound Card (Audio)	Not required for basic operation of E3283S system (Required for E3238S training classes)	32 bit Stereo Audio Card (with Line in/out). (Required for E3238S training classes)
Serial Ports	Not required for basic operation of E3283S system. (Required for Serial handoff receivers)	1-port (Required for Serial handoff receivers)
Parallel Ports	Not required for basic operation of E3283S system	1-port (Use: Parallel printers)
USB	1-port (USB-1 OK) One port required for License Key	4-ports (USB-2) One port required for E3238S License Key
Fire Wire	1-port (Must meet OHCI standard) Required to Connect PC to VXI Mainframe (Daisy chain of other fire wire devices is possible))	2-ports (Must meet OHCI standard) Required to Connect PC to VXI Mainframe (Daisy chain of other fire wire devices is possible))
GP-IB	Not required for basic operation of E3283S system. (Required to control GP-IB handoff receivers)	1-port - (May use LAN to GPIB converter) (Required to control GP-IB *(HP-IB) handoff receivers)
Keyboard / Mouse:	Not required for Laptops. PS/2 or USB required for other controllers (If USB make sure you have enough USB ports)	Not required for Laptops. PS/2 or USB required for other controllers (If USB make sure you have enough USB ports)
Networking	Not required for basic operation of E3283S system. Required for Multiple System Synchronization (MSS) (Use: Connection to other systems on the network.)	100/1000 Mbit/s Network Interface card(NIC). Required for Multiple System Synchronization (MSS) (Use: Connection to other systems on the network.)
PCI Expansion Slots	Maybe required for the above items (Must meet PCI 64 spec for Systran Card) (1U controllers typically only have 1 PCI slot. This may not be enough for other required PCI cards)	One open PCI slot required for (MMS) (Must meet PCI 64 spec for Systran Card) (1U controllers typically only have 1 PCI slot. This may not be enough for other required PCI cards)

Software	MINIMUM	RECOMMENDED
Anti-Virus Software	Not required for basic operation of E3283S system	Symantec Anti-Viruses software
Microsoft Office Suite (Word, Excel & Power Point)	Not required for basic operation of E3283S system (Use: Export of E3238S databases to a spreadsheet for manipulation of data and/or report creation	Microsoft Office Word & Excel (Win2K or XP) (Use: Export of E3238S databases to a spreadsheet for manipulation of data and/or report creation
Microsoft Visual Studio	Not required for basic operation of E3283S system	Visual Studio .net (Use: User Programming (ASD) and Signals Development)
Wind River DIAB Compiler	Not required for basic operation of E3283S system (Use: Required for Signals Development)	DIAB Complier - Node Locked (100-22779-8C) (Use: Required for Signals Development)

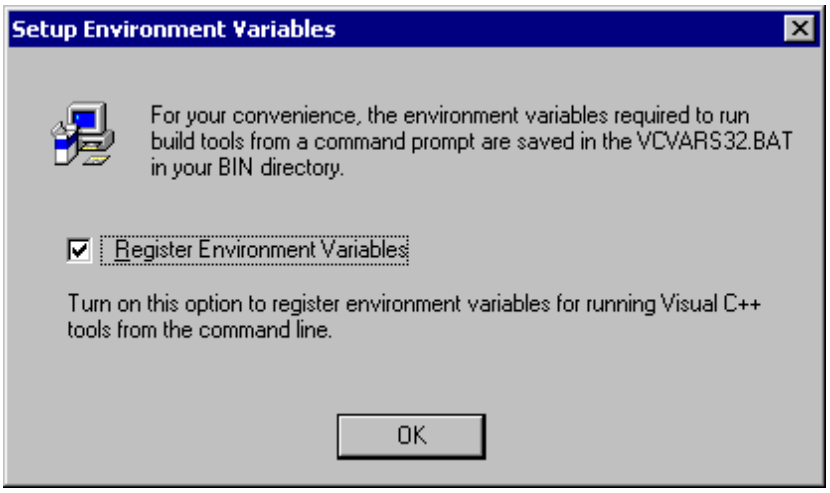
Setting the Compiler Environment Variables for option ASD

The following is required to develop libraries with E3238S option ASD:

- Microsoft Visual Studio Ver 6.0
- Microsoft Visual Studio.NET, Ver 7.0
- Microsoft Visual Studio.NET, Ver 8.0

When installing the Microsoft compiler, we recommend allowing the setup program to register the compiler environment variables necessary for command line compilation. See the following figure.

Figure 43. Visual Studio Installation



If you have already installed Visual Studio 6, Visual Studio.NET Ver 7.0 or Visual Studio.NET Ver 8.0, you may need to run the batch file provided to set the environment variables. This can be found in the Visual Studio Installation path in these locations, depending on your version. The following information assumes the Microsoft Software is installed on the C: drive.

- Visual Studio 6.0 - C:\Program Files\Microsoft Visual Studio\VC98\Bin\vcvars32.bat
- Visual Studio .NET Ver 7.0 - C:\Program Files\Microsoft Visual Studio .NET 2003\Common7\Tools\vsvars32.bat
- Visual Studio .NET Ver 8.0- C:\Program Files\Microsoft Visual Studio .NET 2005\Common8\Tools\vsvars32.bat

Installing and Configuring the E3238S Software

This section describes the procedures used to install the E3238S software and associated libraries for the Windows® 2000/XP operating system on a laptop controller.

Note *None* of the installation procedures given here need be performed on systems delivered direct from the factory. All software on a new system is installed, configured, and tested before it is shipped.

If you have purchased a commodity laptop for your system, you must:

The steps to complete these actions are described in the following procedures.

This section describes the installation and configuration process onto a laptop running the Windows® XP Professional operating system. The procedures are similar on a system running Windows 2000® Server.

Note The E3238S installation will check for necessary IO libraries. If they do not exist, you will be prompted to install the Agilent IO Libraries which will require you to repeat the E3238S installation a second time.

Note Be sure that the E3238S application is installed *before* installing the license software. Also, if your software license uses a USB key, the USB key *must* be disconnected from the computer until after the E3238S application software has been installed.

Configure the Operating System

The first task is to set up the operating system.

1. Click **Start** - **Control Panel** - **Display**.
2. In the Display Properties dialog box, click the **Settings** tab.
3. Set the **Screen resolution** slider to a setting equal to or greater than 1024 by 768 pixels. Note that some windows created by the E3238S program may require a larger screen area than 1024 by 768 pixels. If you encounter this problem, set the slider to a larger screen area.
4. Set the **Color quality** selection to be equal to or greater than 256 colors.
5. Click **OK** to close the Display Properties dialog box. You can also close the Control Panel window.
6. Double-click the desktop icon **My Computer** and then click **Tools**, **Folder Options...**
7. Click the **View** tab
8. Under **Advanced settings,** enable **Show hidden files and folders**.
9. Make sure that **Hide file extensions for known file types** is **disabled**.
10. Click **OK** to close the Folder options dialog box.

Setup the Filesystem

This procedure creates the folders for the E3238S software. Agilent recommends copying an image of the E3238S DVD to the computer's C: drive. Placing the image on the C: drive allows the software to be re-installed without the distribution DVD.

1. On the C: disk, create the folder `C:\Images\E3238S\E3.2`
2. Copy the contents of the E3238S DVD to the folder: `C:\Images\E3238S\E3.2`
3. If you are installing any options from a separate CD or DVD, create the folder: `C:\Images\E3238S\E3.2\`*Option*

   Where *Option* is the name of the software of the option you will install.
4. Copy any option CD or DVD to the appropriate folders.

   Installing the options is described in Installing Software Options (page 73)

Install the E3238S Program

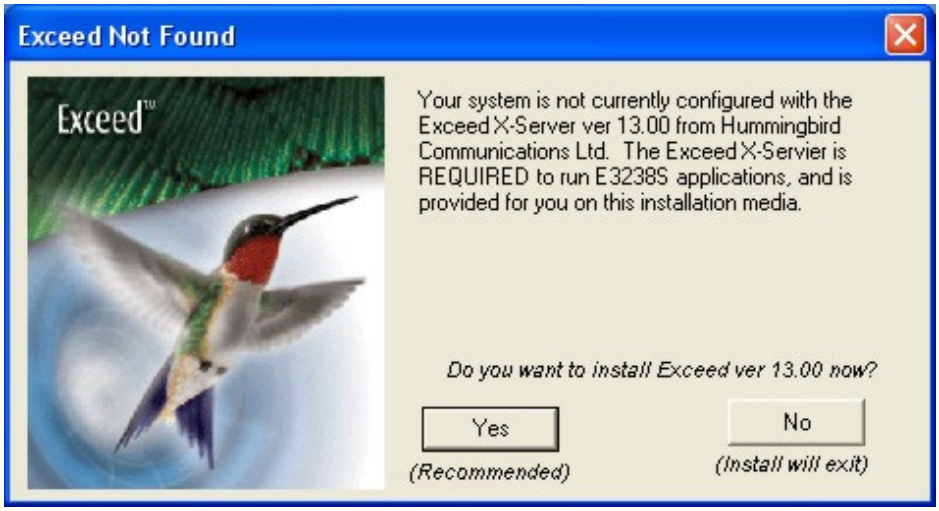
Note

Be sure that the E3238S application is installed *before* installing the license software. Also, if your software license uses a USB key, the USB key *must* be disconnected from the computer until after the E3238S application software has been installed.

1. Insert the Agilent E3238S application DVD in the drive. It should auto-start and display an opening screen. You may also choose to run the installation from `C:\Images\E3238S\E3.2\Winnt\Setup.exe`
2. From the Installation screen, click "`Install 35688E now`," read the next screen and click `Next`.
3. Select your hardware platform. The instructions that follow are for the VXI hardware platform. Please refer to the installation documentation for your hardware platform if you select anything other than VXI.
4. If no IO Libraries are installed, you will be prompted to install the Agilent IO Libraries. You have 2 options, install the Agilent IO Libraries or Exit Setup to install a third party driver.

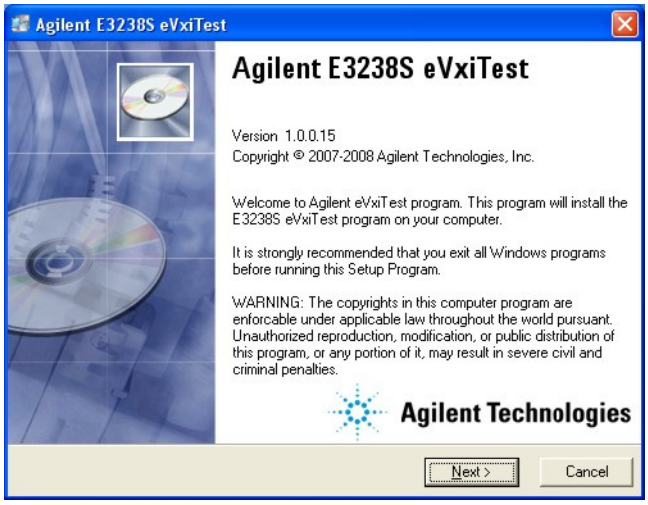
5. Once you have selected to install the Agilent IO Libraries, you will be given a notice that you must Restart the installation of the E3238S software.

6. When the Agilent IO Libraries installation is complete, clear the "`Yes, I want to connect to my instrument. Launch Agilent Connection Expert`" selection. Click `Finish` and the E3238S installation should resume. If it doesn't, re-start the E3238S installation following the instructions in step 1.
7. If the Exceed X Server is not present on your system, you will be prompted to install the Exceed X server.[1]



8. Click "Yes" to install the Exceed X Server Software
9. Once the Exceed installation is complete, the E3238S installation should continue automatically. If it doesn't, re-start the E3238S installation following the instructions in step 1.
10. Click "Finish" to complete the E3238S installation
11. The E3238S installation will then provide a simple VXI test called eVxiTest to check that your hardware is accessible. It is recommended that you install this test.

[1]If the XP Firewall is enabled, the Exceed installation will generate a security alert.



12. Click "Finish" to complete the installation. You will be prompted to re-start.

13. When the installation is completed, remove the E3238S installation DVD.

Configuring the VXI interface

To configure your VXI interface:

1. Connect the IEEE 1394 interface card from the computer to the E8491 VXI module in the mainframe. Power on the computer and the VXI mainframe.

Note

If the IEEE 1394 adapter card was just installed, the Found New Hardware Wizard may open. If the Found New Hardware Wizard opens, click **Next** and when it is finished, click **Finish**.

2. Run Connection Expert:

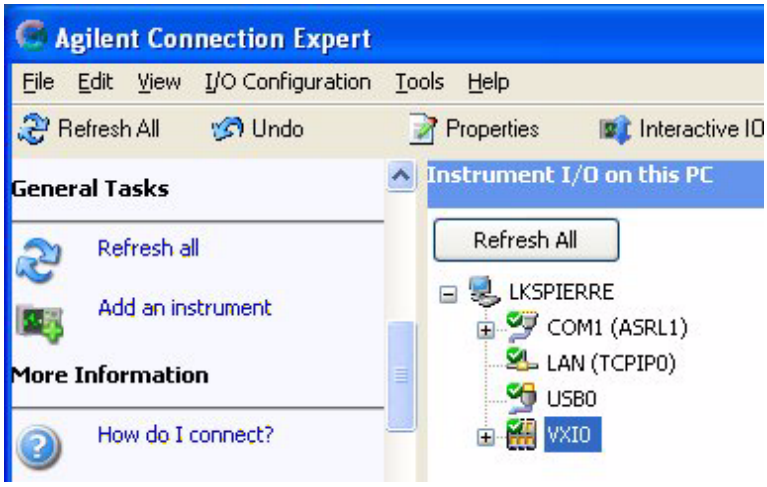
   Click the IO Control icon in the task bar, then click **Agilent Connection Expert**. (Or click **Start > (All) Programs > Agilent IO Libraries Suite > Agilent Connection Expert.)**

   Connection Expert will automatically detect and configure most interfaces and instruments and will assign names and other default configuration settings.

   If the VXI0 interface does not show up, check cable connections and power cord connections. Also, check another port on the E8491 VXI Interconnect Module.

   Click **Refresh All** to update.

Figure 44. IO Lib Menu



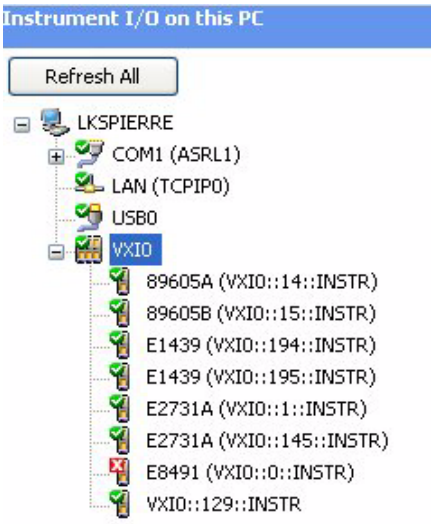
Note

If Connection Expert shows a VXI0 and VXI1 interface, delete both of them and click **Refresh All**. This will remove the VXI1 interface. Connection Expert should only show the VXI0 interface.

Interfaces or instruments can also be added manually to the test system configuration. See Agilent IO Libraries Suite Online Help for more information.

3. To verify that the computer recognized all the modules in your VXI mainframe, expand **VXI0** in the *Instrument I/O on this PC* section. Connection Expert will list all the modules and their addresses. A green check mark is shown to verify the VXI mainframe and modules are on and connected properly. The E8491 will always show a red x.

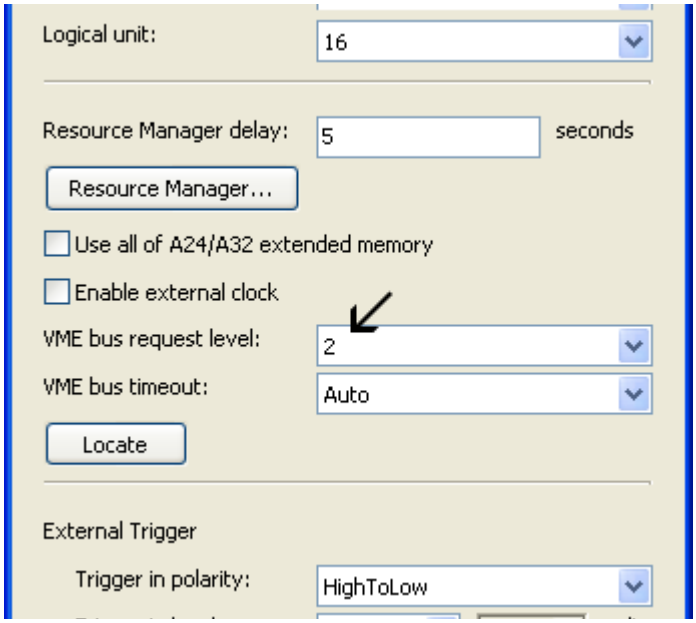
Figure 45. IO Config



4. In order to use the E3238S Software with your configured IO Devices, you must change the VME Bus Request level to 2. The Agilent IO Library default value is 3. To change the VME Bus Request:

- Make sure the VXI0 is selected (highlighted), then Right Mouse click and select "Change Properties"

Figure 46. VME Bus Level



Make sure the "VME bus request level" is set to "2". Then select OK to save your settings.

Agilent IO Libraries Suite also provides a VXI Resource Manager. This is a software utility that initializes and prepares a VXI system for use. The VXI Resource Manager runs when any of the following conditions occur:

- You start it from the Connection Expert's Tools menu (select **Tools > VXI Resource Manager > Edit Resources**, then click **Run** in the resulting Resource Manager dialog box)
- You start it from the Agilent IO Control
- You apply or cycle VXI mainframe power
- You press the E8491 Reset button
- You reboot your PC

In VXI systems with multiple E8491 interfaces, you can turn off individual VXI mainframes without affecting other mainframes in the system. When a mainframe is turned on, the VXI Resource Manager re configures that mainframe.

For more information about VXI interfaces, see Connectivity in the Documentation section of the Agilent IO Libraries Suite. You can find this when you click on the **IO Control** icon in the task bar and then select **Documentation**. (Or click **Start > (All) Programs > Agilent IO Libraries Suite > Documentation.)**

Installing Software Options

If your system has one or more software options, use the following procedure to install and configure them:

1. If you have not yet done so, copy the library DVD to the system disk.

   We recommend copying the contents of the E3238S DVD onto the system disk to: `C:\Images\E3238S\E3.2\`*Option*

   Where *Option* is the name of the software of the option you will install.

2. Run the installation program by executing the `setup.exe` program in the directory: `\images\E3238s\E3.2\`*Option*`\`

3. Modify the `e3238s.cfg` file to enable the selected libraries:

   The sections of the `e3238s.cfg` file that control the libraries are found at the file's end. Details are given in a readme.txt file installed in each library's install directory (e.g., narrow band recorder, option NBR, is installed in `\E3238s\NBR`)

   To enable a library,

   a. Using a text editor, open the configuration file `\E3238s\e3238s.cfg.`

   b. Scroll to the bottom of the file

   c. Remove the exclamation points preceding the appropriate command lines controlling the desired library.

   d. Enter the appropriate arguments to the setup commands.

   Follow the instructions given in the ReadMe file to properly configure the option settings (e.g., maxChannels, loadFactor, etc.)

4. The ReadMe file may also include information describing application settings that must be used to ensure the proper operation of the option. See the following example.

In the following example the Direction Finding library and the Modulation Recognition library are enabled.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!                       Direction Finding                                      !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
df1.enabled:           True
df1.hostLib:          c:/e3238s/DF1/DF1.dll
!df1.args:
!df1.alias:
!df1.latitude:
!df1.longitude:
!df1.declination:
!df1.heading:


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
!                     Modulation Recognition                                   !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!
modRec1.enabled:        True
modRec1.hostLib:       c:/e3238s/MR1/MR1.dll
!modRec1.args:
!modRec1.alias:
```

Software Licensing

The E3238S software requires a license file to run. This section describes where this file is located, what it is named, and how to resolve some common license problems.

To run the E3238S software, two licensing elements must be in place:

- A valid license file
- A device that supplies a hostID for the license

The hostID can be supplied by either of the following:

- A specific computer
- A USB Key

The license file is provided by Agilent on the License File CD. It is an ASCII file which contains the license number, specifies the valid options, and identifies the activation key.

Caution Changing the license file invalidates it and disables the application. The file may be displayed but should not be modified. When changes are necessary, such as for version updates or added options, a new file will be issued.

License Installation

Notes Be sure that the E3238S application is installed *before* installing the license software. Also, if your software license uses a USB key, the USB key *must* be disconnected from the computer until after the E3238S application software has been installed.

The first time the USB Key is connected to a USB port, Windows runs the New Hardware Found wizard. This wizard associates the USB Key with the correct software drivers.

From a License CD If you have received a license file CD, then install the E3238S software license file as follows:

1. Place the E3238S license CD into the system's CD or DVD drive.
2. When the automatic installation procedure displays its window, click the button labeled "Next."

When the license installation is complete, remove the license CD and store it in a safe place.

From a File If you have received a license file in some other manner, such as by E-mail, and the E3238S software is installed on the computer, install the license file as follows:

1. Copy the license file to the `\E3238s\licenses\new` directory.
2. Run the license installer program that was installed with the E3238S program. Click **Start - Programs - Agilent E3238S - Tools - License Installer**

When finished, save a copy of the original license file in a safe place.

Common problems and their solutions are listed on the next page.

Dealing with License Problems

If the E3238S software is unable to validate its license, an error message is displayed. If you experience license problems, check for the following common causes.

- **Is the license file at the location expected by the E3238S software?**
  Check for the presence of the environment variables
  `AGILSURV_LICENSE_FILE` or **`LM_LICENSE_FILE`**.
  These are used to specify the directory that contains the license file, (typically C:\E3238s\license\).

  There are two ways to check the environment variables:

  - Open a command window and type "set" (and Enter) at the prompt.
  - Right-click My Computer (icon), click Properties , Advanced (tab), Environment Variables (button)

  Operational license file names end with the .lic extension.

- **Is the license current?**
  Check that the license name does not end in -temp.lic. If it does, it is a temporary license and may have expired. To see if the license has expired, open the file with a text editor; the expiration date is shown on the INCREMENT line. See the note below.

- **Is the license valid for the hardware?**
  - If the license is for a USB Key, make sure that you have the correct USB Key connected securely in the computer's USB port. Compare the serial number listed in the license file with the number on the USB Key.
  - If the license is keyed to a specific computer, make sure that the correct computer is being used. To do this:
    1. Run the HostID utility (Start - Programs - Agilent E3238S - HostID). This displays the computer's host ID. (pairs of numbers separated by colons)
    2. Open the license file in a text editor. The same number sequence should follow HOSTID=

Note The E3238S application is enabled only when the license file and hostID is as provided by Agilent. If changed, the hostID string can be changed back to the proper value and continue to function properly.

- **Is the license valid for the current version of the software?**
  Licenses for earlier versions of the E3238S software may not be valid for later releases. See the note below.

- **Is the license valid for your software options?**
  Make sure that the license directory contains a valid license. It is acceptable to have more than one license file in the license directory. This is useful for multiple systems and multiple USB Keys. See the following note.

Note Old license files can cause problems. You should rename (change the extension or add another extension after .lic) or delete all old license files.

The Software License File

The E3238S software requires a license file to run. This section describes where this file is located, what it is named, and how to resolve some common license problems.

For new, Factory Integrated systems

If your system was integrated at the factory, your license file is correctly configured for using your system. Place the License File CD in a secure location.

For Other Users

This license file is provided directly by Agilent; it is unique for each installation. The license file is installed using the License File CD which is distributed by Agilent for the software installation procedure. The license file is valid only for the software configuration that you have purchased. The license file cannot be edited and remain valid.

If any parameter of the license must be changed (for example, addition of new libraries, expiration date, version number, Ethernet address, or USB Key hostID) the license file must be re-generated by Agilent. The new license must then be re-installed.

The E3238S software runs only on a controller or PC upon which its license is valid. The software checks that it is running on a proper controller by one of two processes:

- Verifying the presence of a USB Key in a USB port. This is the most flexible method. This option allows the software to be installed and run on various PCs by moving the key to the new PC.
- Verifying the computer's hostID. This option locks the software to one host computer. This is the simplest method when the software can be locked to a single computer.

Note

The software runs only when the license file is properly installed and the corresponding hostID is present. A computer-keyed license locks the software to a specific computer. A USB-keyed license locks the software to a specific USB Key.

The E3238S software and license file for the USB Key may be installed on several different computers. If this is the case, then to use any specific computer, you need only connect one of the valid USB Keys to that computer's USB port.

License File Location

During the installation of the E3238S software, the installation process creates an environment variable named AGILSURV_LICENSE_FILE; this variable specifies the pathname of the license file. By default this location value is:

```
\E3238s\licenses\
```

If the environment variable AGILSURV_LICENSE_FILE does not exist, or if the software cannot find the license file at the pathname specified by the variable, the software checks for the presence of a second variable, LM_LICENSE_FILE. Note that this second environment variable is not automatically created and has no default value.

The value of either variable can be changed to modify the location and name of the license file, but use care to ensure that the contents of one of the variables match the pathname of the valid license file.

If the E3238S software is, at startup, unable to validate its license because the license file is not in its expected location or the license does not match the expected controller hardware, the software displays an error message. For more information on resolving license problems, see Licensing Validation Problems (page 81).

Temporary Licenses

If a copy of the E3238S software is used for training or demonstration purposes, it may be provided with a temporary license. Temporary licenses can be identified by the suffix `-temp.lic` in their names. When a long-term license is installed, the temporary license should be deleted or re-named with a filename extension other than .lic. This action prevents the temporary license from interfering with the E3238S software recognizing its correct license.

If a temporary license expires while the E3238S software is running, the software provides a 10-minute grace period to save mission data and setup files before shutting down.

USB-Keyed Licenses

A USB-keyed license can be installed on any PC that runs the E3238S software (E.01.00 version or higher) and has a free USB port. This license allows the software to start if the corresponding key is connected to one of the PC's USB ports and the value of the license file matches the USB Key hostID. If the key is not present when the software starts, the software displays an error message.

USB Key-based license files have names like the following example:

```
N6820E-20JW1234-FLEXID=9-7e1234e6-SWL-14Apr2006.lic
N6820E-20JW1234-FLEXID=9-7e1234e6-SWL-14Apr2006-temp.lic
```

Where `20JW1234` is the license number, `9-7e1234e6` is the FLEXid (USB Key's hostID), and the date shown is the creation date.

The first time the USB Key is inserted in a USB port, Microsoft Windows runs the 'New Hardware Found' wizard. This wizard associates the hardware key with the correct software drivers. After this first insertion, Windows will run the New Hardware Found wizard only when the key is installed in a different USB port.

If the key is removed while the software is running, within a few minutes the software displays an error message. This message notifies the user that the license that allows the E3238S software to run is no longer valid. When this message is displayed, the software provides a 10 minute grace period to save mission data and setup files before shutting down. Good practice recommends immediately saving your mission data and setup files, then re-attaching the USB Key and clicking the dialog box's "OK" button.

Computer-Keyed Licenses

The license file is installed on the PC that has the corresponding hostID. This value is composed of information derived from the computer.

If the hostID does not match the computer when the E3238S software starts, the software displays an error message.

The E3238S GUI interface will halt until the component is returned and the user clicks the dialog box's "OK" button.

License files provided by Agilent have names like the following example:

```
N6820E-20JW1234-11ab56789a12-SWL-14Apr2006.lic
N6820E-20JW1234-11ab56789a12-SWL-14Apr2006-temp.lic
```

Where `20JW1234` is the license number, `11ab56789a12` is the hostID, and the date shown is the creation date.

In Case of Failure

This recommendation assumes that you have a failure recovery plan for the entire system that includes spares for vital components, especially the computer. To recover from a computer failure, a backup computer should be available that has the software installed and is properly configured. The following discussion describes how to activate the software license on the backup system.

Computer Failure

If the computer has failed and a backup computer is available:

- **On a USB-keyed system**: move the USB Key to the backup system and continue operation.
- **On a computer-keyed system**: (see below)

License hostID Device Failure

This section describes how to recover from the loss of a functional license due to a hardware failure in your hostID device. This is indicated by an error message stating that the E3238S licensing system could not validate the license.

Each of the two licensing methods has its associated failure mechanism.

- USB-key based licensing relies on the presence of the USB Key. The failure or loss of the original USB Key disables the licensing on this kind of system.
- Computer-key based licensing relies on a computer identification which is derived from its hardware components. The failure or removal of the components used to characterize it will disable the licensing on such a system.

USB-key Recovery

If the USB Key fails, notify the E3238S license administration team. See Licensing Support (page 81). Arrangements will be made to send a replacement license file and USB key. You will be asked to return the failed USB key to Agilent.

To continue operation until the replacements arrive, you have two choices:

1. The license administration team can E-mail you a temporary computer-keyed license file to enable a particular system to run.
2. Install the Backup Key. See The Backup Key (page 80).

Computer-key Recovery

Notify the E3238S license administration team of the failure. See Licensing Support (page 81). Arrangements will be made to send a replacement computer-keyed license file. (This can be done via E-mail.) You will be asked to provide documentation of the computer's failure.

To continue operation until the replacement file arrives you can install the Backup Key. See The Backup Key (page 80).

The Backup Key

To assure a quick, fail-safe recovery mechanism for the software licensing, a Backup Key is provided. This special USB Key is shipped in a package marked as a "**single-use emergency backup key**."

The Backup Key will enable operation of the E3238S software for a limited time[1] whether the original key mechanism was USB-key based or computer-key based. This provides temporary license recovery for situations where you are unable to contact the license administration team at the time of the failure and when you need your system to be back up and running in a very short period of time.

[1]The length of time that the backup key will activate the program is specified on a notice that is delivered with the key.

Licensing Validation Problems

If you are experiencing problems with your license, check for the following common causes.

Is the license file at the location expected by the E3238S software?
Check for the presence of the environment variables AGILSURV_LICENSE_FILE and LM_LICENSE_FILE. If either of them are present, make sure that they describe the path that contains the license file, (typically `\E3238s\licenses`). Make sure that directory contains a valid license and that the license file name ends in ".lic".

Is the license current?
If the license name ends in -temp.lic, it may have expired. To check whether a temporary license has expired, open the license file using a text editor and find the first date that appears in the file. This is the license's expiration date.

Is the license valid for the hardware?
On computer-keyed systems, make sure that the license file's hostID (shown both in the file name and in the file text) matches one of the hostID values found in the file C:\temp\e3238sHostID.text file written by the HostID utility program. To run this utility program, click **Start - Programs - Agilent E3238S - HostID**

When the license is locked to a USB Key, make sure that you have the correct key plugged securely into the laptop's USB port.

Is the license valid for the current version of the software?
Licenses for earlier versions of the E3238S software may not be valid for later releases.

Licensing Support

To contact Agilent regarding licensing needs:

Agilent E3238S Software Licensing Administration:

Telephone:	(425) 356-6261
E-mail:	eveswl@agilent.com
Fax:	(425) 356-6260
Hours:	8 AM to 5 PM Pacific Time, Monday-Friday except Agilent holidays.

Temporary and replacement files require information created with the HostID utility. Start - Programs - Agilent E3238S - HostID. This creates the file `C:\temp\e3238sHostID.txt` which can be emailed to Agilent.

Modifying the Hardware Configuration File

The E3238S application obtains hardware configuration information from the file `e3238s.cfg`, which is located in the E3238S directory (e.g., `C:\E3238s`). A default hardware configuration file (`d.e3238s.cfg`) is also installed in this directory. If an `e3238s.cfg` file is not found when the E3238S software is installed, the default configuration file is automatically copied to `e3238s.cfg`.

The default hardware configuration specifies an E1439D ADC at logical address 130, and a E9821A DSP at address 128. If your hardware setup does not match this configuration, you must modify the `e3238s.cfg` file.

For more information about hardware configuration and the definitions used in the `e3238s.cfg` file see Hardware Installation on page 7 and Hardware Configuration Reference on page 89.

Upgrades

When upgrading E3238S software, new features may not work properly until the new configuration information is specified in the "old" `e3238s.cfg` file. If you are installing the upgrade over the previous version, (i.e., the `e3238s.cfg` configuration file already exists in the E3238S directory) the configuration file is not overwritten. This avoids losing the information in the existing file.

To setup the new configuration

1. Rename the existing `e3238s.cfg` if you wish to save it
2. Copy the file `d.e3238s.cfg` to `e3238s.cfg`
3. Edit the new `e3238s.cfg` file so that it accurately defines the hardware configuration

You can use the eVXItest application to help determine what hardware is installed.

Configuring X Window Application Resources

The X Windows system uses a *resource* file to allow users to control various application variables. Those created specifically for the E3238S are described in the section called "Application Resources" on page 183.

A default resource file for the E3238S program (`d.E3238s`) is provided in the E3238s directory (e.g., `C:\E3238s`). To modify E3238S resources, copy the `d.E3238s` file to `E3238s` and edit the file as appropriate.

To customize the resources for a specific user, place the modified E3238s file in the user's `Profile` directory (`C:\WINNT\Profiles\<user_name>`), or in the user's HOME directory as defined in the Windows® User Manager.

The resource file in the E3238S directory (e.g., `C:\E3238s`) applies to all users that do not have a resource file in their Profile or HOME directories.

When the application is started, the window manager searches a number of places until it finds the E3238s resource file. It uses the first one it finds. The resource file search order is as follows:

1. User Profile directory
2. User HOME directory
3. E3238S directory (as defined during installation)
4. Standard Exceed locations (XUSERFILESEARCHPATH env var & others)

See the Exceed X *Development Kit User's Guide* for more information.

Miscellaneous E3238S Configurations

Secure Display Setup

The E3238S application contains a security feature that blanks the E3238S window when you select **Utilities**, **Secure Display** (or press Ctrl-S).

- By default, this feature is enabled. It may be disabled by editing the e3238s.cfg file and commenting out the line `disableAccess: Secure Display`
- To restore the normal application display, enter the current user's password.
- To determine the current user's name, press Ctrl-Alt-Del.

Using a blank password

Windows XP security policy can interfere with the use of no password in locations other than the main console logon. To resolve this issue, disable it as follows:

Control Panel...
Administrative Tools...
Local Security Policy...
Local Policies...
Security Options...

If the entry listed below is listed as 'Enabled', double-click it and select 'Disabled':

Accounts: Limit local account use of blank passwords to console logon only:

When this is disabled, user accounts that have no password can recover the secured display without entering a password.

Access Control Security

The E3238S application supports control of access to entries in the main menu bar and popup menus by a system administrator. See the dialog box called by **File, Access Control**.

Only the items enabled in this dialog box may be accessed by users. This feature is password protected using a special Windows® user account named `e3238s`.

- When this account exists, its password must be used to make access changes.
- When this account does not exist, no restrictions exist for changing access.

To view existing accounts or setup a new one see **Start, Run and enter "compmgmt.msc". Select the "+" sign next to "Local Users and Groups"**.

Printer Configuration

The E3238S prints to any printer currently defined on the Windows workstation; see **Start, Printers and Faxes**. To automatically specify a printer other than the default or to set other print options (and avoid displaying the print dialog box) see the discussion in the `d.print` file in the E3238S directory.

File System Organization

The application executables are in the `$E3238S\bin` directory.

The product documentation files are in `$E3238S\manuals` (PDF files).

The E3238S software license file(s) are in the `$E3238S\license` directory.

Error correction files are in `$E3238S\cal`.

Optional applications each have their own directory. Examples are as follows:

```
$E3238S\ctcss           (CTCSS[1] recognition and recording, option PL1)
$E3238S\fm              (FM signal recognition and recording, option FMR)
$E3238S\mr1             (modulation recognition, type 1 - wide band)
$E3238S\vad             (voice activity detector, HF)
$E3238S\uvad            (voice activity detector, VHF/UHF)
$E3238S\pager           (pager intercept)
$E3238S\audio           (audio output; uses DDC channels & PC audio output)
$E3238S\featureStudio   (used to develop feature extraction & energy filter libs)
$E3238S\nbr             (narrow-band recorder)
```

Network Services

To support networked communication between the E3238S application and external socket programs, there must be an entry in the `C:\WINNT\system32\drivers\etc\services` file as follows:

```
e3238s    7011/tcp
```

This defines a sockets port and is usually added during installation. This entry must appear in the services file of both the workstation running E3238S and the remote workstation (the numbers must be the same). Verify that the entry exists as shown above. If not, add it.

[1]Continuous Tone Coded Squelch System (PL is for 'private line', a misnomer)

User Programming

This section describes how to configure the system to use the User Programming feature, option ASD. This procedure establishes access to User Programming features for a single user as well as creating directories for shared library development.

On a Microsoft Windows® system, the application files may be installed wherever the user indicates. The default location is the `C:\E3238s` directory but since this may vary from one installation to the next, the label `$E3238S` is used to indicate the application's home directory in the following discussion.

The include and shared library files are installed in the following directories:

```
$E3238S\include      include files
$E3238S\lib          shared library files
```

The User Programming example source files are organized under the examples directory. Each User Programming shared library type has its own Makefile and directory as follows:

```
$E3238S\examples\alarmTasks
$E3238S\examples\fileFormats
$E3238S\examples\filterAndFeatures
$E3238S\examples\genericLib
$E3238S\examples\handoffReceiverDriver
$E3238S\examples\sockets
$E3238S\examples\userMenu
$E3238S\examples\userPane
```

Creating a Development Environment

This discussion describes how to create the user programming development environment for a single user. The home directory for this user is represented as <HOME>. In the following steps, replace <HOME> with the full pathname of the user's home directory.

1. Create the ASD development directory under your <HOME> directory.

```
cd <HOME>
md asd
```

2. Install and modify your personal copy of the E3238S configuration file.

```
copy <E3238S>\e3238s.cfg <HOME>\e3238s.cfg
```

Edit the `e3238s.cfg` configuration file to enable the socket server and add the existing shared libraries:

```
energyHistoryFilter:  <HOME>\asd\filterAndFeatures\filterAGE.dll,\
                      <HOME>\asd\filterAndFeatures\filterTEST.dll

userAlarmTask:        <HOME>\asd\alarmTasks\demoUserTask.dll

featureExtraction:    <HOME>\asd\filterAndFeatures\featurePWR.dll,\
                      <HOME>\asd\filterAndFeatures\featureDF.dll

userMenu:             <HOME>\asd\userMenu\userMenu.dll,\
                      <HOME>\asd\userMenu\userMenuCmnd.dll,\
                      <HOME>\asd\userMenu\userMenuArrow.dll

userPane:             <HOME>\asd\userPane\userPane.dll

genericLib:           <HOME>\asd\genericLib\genericLib.dll
```

3. Install a personal copy of the E3238S application resource file and modify it.

```
copy <E3238S>\d.E3238s <HOME>\E3238s
```

Note that this copies the file and renames it.

Edit the resource file and add the following line specifying the hardware configuration file to load when the application starts:

```
*hardwareConfiguration: <HOME>\e3238s.cfg
```

If you need multiple lines in the toolbar due to adding user-defined menus, add/modify the following resources to read:

```
*toolbarWrap: True
*toolbar.paneMaximum: 170
```

For more information about E3238S application resources see pg 183.

4. Copy the example files (source files) to your private development directory:

```
cd $Home\asd

copy /s $E3238S\examples\alarmTasks
copy /s $E3238S\examples\fileFormats
copy /s $E3238S\examples\filterAndFeatures
copy /s $E3238S\examples\genericLib
copy /s $E3238S\examples\handoffReceiverDriver
copy /s $E3238S\examples\sockets
copy /s $E3238S\examples\userMenu
copy /s $E3238S\examples\userPane
```

5. Rebuild the object files (dynamic libraries) from the source in your private development directory.

Hardware Configuration Reference

This section describes the hardware parameter settings used in the initialization configuration file typically named `e3238s.cfg`. This file defines the system hardware configuration and is loaded when you start the `e3238s` program. If the information in this file does not match the installed configuration, error messages are displayed to help isolate the problem.

Note This file is installed as `d.e3238s.cfg` during upgrades or reinstallation to avoid overwriting any existing configuration information. Upgrades may contain new commands to support new features so some manual editing may be required before running the application program.

Note Some software options, such as option USD, provide their own configuration file that can be cut and pasted into the e3238s.cfg file. The configuration files are located in the directory for the specific option. For example, the USD configuration file default location is `C:\E3238s\usd\d.e3238s.cfg`.

Note The N6841A RF Sensor has its own configuration file. Refer to the documentation that came with the N6841A RF Sensor for configuration information.

Parameter List

asxDsp_0

Syntax `asxDsp_0: filename`

`The maximum length is 127 characters.`

Description Specifies the file that provides the ability to load and execute more than one signal processing library to be downloaded to the first G4 on a dual-G4 processor PMC card.

elf = Executable and Linking Format
esl = elf shared library

Example

```
multipleSignalsPerProcessor: Enabled

asxDsp_0:   C:\E3238s\downloadables\ASXdsp_0.elf
asxDsp_1:   C:\E3238s\downloadables\ASXdsp_1.elf

!
! Demo Signal Type
!

signal1.enabled:     False
signal1.hostLib:     C:\E3238S\demo\demoHost.dll
signal1.hostDsp:     C:\E3238S\demo\demoDsp.dll
signal1.targetDsp:   C:\E3238S\demo\demoDsp.esl
signal1.loadFactor:  64
signal1.minChannels: 4
signal1.maxChannels: 32
signal1.args:
signal1.alias:
```

Note You may use either regular slashes (/) or back slashes (\) in the pathname.

See Also multipleSignalsPerProcessor (page 132)

signal.targetDsp (page 171)

asxDsp_1

Syntax `asxDsp_1: filename`

`The maximum length is 127 characters.`

Description Specifies the file that provides the ability to load and execute more than one signal processing library to be downloaded to the second G4 on a dual-G4 processor PMC card.

elf = Executable and Linking Format
esl = elf shared library

Example

```
multipleSignalsPerProcessor: Enabled

asxDsp_0:   C:\E3238s\downloadables\ASXdsp_0.elf
asxDsp_1:   C:\E3238s\downloadables\ASXdsp_1.elf

!
! Demo Signal Type
!

signal1.enabled:     False
signal1.hostLib:     C:\E3238S\demo\demoHost.dll
signal1.hostDsp:     C:\E3238S\demo\demoDsp.dll
signal1.targetDsp:   C:\E3238S\demo\demoDsp.esl
signal1.loadFactor:  64
signal1.minChannels: 4
signal1.maxChannels: 32
signal1.args:
signal1.alias:
```

Note You may use either regular slashes (/) or back slashes (\) in the pathname.

See Also multipleSignalsPerProcessor (page 132)

signal.targetDsp (page 171)

df.alias (EDF)

Syntax `df(1..8).alias: string`

`The maximum length is 31 characters.`

Description Specifies an alternate name to be used in the application's user interface so that the real signal name is not displayed.

Example

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                          Direction Finding                                   !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

df1.enabled:          False
df1.hostLib:          C:\E3238s\lib\demoDF.dll
df1.args:
df1.alias:            NorthPole
df1.latitude:
df1.longitude:
df1.declination:
df1.heading:
```

Note You may use either regular slashes (/) or back slashes (\) in the pathname.

df.args (EDF)

Syntax `df(1..8).args: string`

`The maximum length is 255 characters.`

Description Specifies the values for any parameters that a custom library might pass in.

Example

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                          Direction Finding                                  !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

df1.enabled:          False
df1.hostLib:          C:\E3238s\lib\demoDF.dll
df1.args:             45 90 135 180 360
df1.alias:            NorthPole
df1.latitude:
df1.longitude:
df1.declination:
df1.heading:
```

Note You may use either regular slashes (/) or back slashes (\) in the pathname.

df.declination (EDF)

Syntax

```
df(1..8).declination: float
```

-180 ≤ *float* ≤ 180

Description

Specifies the initial declination at the system location.

Magnetic declination (or magnetic variation) at any point on the earth is an angle that must be added to or subtracted from a compass reading to derive true North.

Values are entered as decimal values. This parameter may be changed in the GUI.

Example

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                       Direction Finding                              !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

df1.enabled:         False
df1.hostLib:         C:\E3238s\lib\demoDF.dll
df1.args:
df1.alias:           NorthPole
df1.latitude:        47.21
df1.longitude:       75.42
df1.declination:     -15.2
df1.heading:         80.0
```

See Also

df.heading (EDF) (page 98)
df.latitude (EDF) (page 100)
df.longitude (EDF) (page 101)

Note

This parameter is saved as part of the mission state in the current group. If you recall a mission state of a different location, this parameter value may be incorrect.

df.enabled (EDF)

Syntax

`df(1..8).enabled:` *`enum`*

`enum` `= {False, True}`

Description

Specifies whether the direction finding components are to be loaded. This allows you to disable a signal library without commenting out all the lines associated with it.

Another way to disable a signal library is to leave the setting True and just comment out the signal.enabled line.

Example

The following example shows a DF library that is disabled:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                       Direction Finding                              !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

df1.enabled:          False
df1.hostLib:          C:\E3238s\lib\demoDF.dll
df1.args:
df1.alias:            NorthPole
df1.latitude:         47.21
df1.longitude:        75.42
df1.declination:      -15.2
df1.heading:          80.0
```

Note

You may use either regular slashes (/) or back slashes (\) in the pathname.

df.heading (EDF)

Syntax

`df(1..8).heading: float`

`0 ≤ float ≤ 360`

Description

Specifies the initial heading of the DF antenna as taken from a compass reading.

This value can be modified by the declination setting (pg 96) to derive the true heading with respect to true North.

Values are entered as decimal values. This parameter be changed in the GUI.

Example

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                       Direction Finding                              !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

df1.enabled:         False
df1.hostLib:         C:\E3238s\lib\demoDF.dll
df1.args:
df1.alias:           NorthPole
df1.latitude:        47.21
df1.longitude:       75.42
df1.declination:     -15.2
df1.heading:         80.0
```

See Also

df.declination (EDF) (page 96)
df.latitude (EDF) (page 100)
df.longitude (EDF) (page 101)

Note

This parameter is saved as part of the mission state in the current group. If you recall a mission state of a different location, this parameter value may be incorrect.

df.hostLib (EDF)

Syntax `df(1..8).hostLib:` *`filename`*

`The maximum length is 127 characters.`

Description Specifies the filename of the DF library to be loaded in the host (system controller). This provides the host component of the typical host-target interaction mechanism.

Example

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                       Direction Finding                             !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

df1.enabled:          False
df1.hostLib:          C:\E3238s\lib\demoDF.dll
df1.args:
df1.alias:            NorthPole
df1.latitude:
df1.longitude:
df1.declination:
df1.heading:
```

Note You may use either regular slashes (/) or back slashes (\) in the pathname.

df.latitude (EDF)

Syntax

`df(1..8).latitude: float`

`-90 ≤ float ≤ 90`

Description

Specifies the initial latitude position of the system.

Values are entered as decimal values.
Negative values indicate a location in the Southern Hemisphere.
This may be changed in the GUI.

Example

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                       Direction Finding                              !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

df1.enabled:          False
df1.hostLib:          C:\E3238s\lib\demoDF.dll
df1.args:
df1.alias:            NorthPole
df1.latitude:         47.21
df1.longitude:        72.45
df1.declination:
df1.heading:
```

See Also

df.declination (EDF) (page 96)
df.heading (EDF) (page 98)
df.longitude (EDF) (page 101)

Note

This parameter is saved as part of the mission state in the current group. If you recall a mission state of a different location, this parameter value may be incorrect.

df.longitude (EDF)

Syntax `df(1..8).longitude: float`

`-180 ≤ float ≤ 180`

Description Specifies the initial longitude location of the system. This may be changed in the GUI.

Values are entered as decimal values. Negative values indicate the Western Hemisphere.

Example

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                       Direction Finding                             !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

df1.enabled:          False
df1.hostLib:          C:\E3238s\lib\demoDF.dll
df1.args:
df1.alias:            NorthPole
df1.latitude:         47.21
df1.longitude:        75.42
df1.declination:
df1.heading:
```

See Also df.declination (EDF) (page 96)
df.heading (EDF) (page 98)
df.latitude (EDF) (page 100)

Note This parameter is saved as part of the mission state in the current group. If you recall a mission state of a different location, this parameter value may be incorrect.

disableAccess

Syntax `disableAccess: string`

`The maximum length is 255 characters.`

Description Specifies which items in the menu bar pulldown menus are inactive at startup. This allows the control of access to the main menu features.

The argument is a string containing the name of the menu label exactly as it appears in the in the pulldown menu. See the following example. Note that multiple strings passed to this command are not delimited by commas. Only spaces separate the values.

Example The following commands disable user control of one entry in the File menu and all entries in the Edit menu:

```
disableAccess: Secure Display
disableAccess: Log Files ...
disableAccess: Clear Log
disableAccess: Clear Log File
disableAccess: Clear Energy History
disableAccess: Clear Signal Database
disableAccess: Clear Frequency Lists
disableAccess: Clear Audio Output
disableAccess: Clear All
```

The user may change the menu access status by entering the password for user `e3238s` when prompted.

i. Menu access control exists in the GUI in the File, Access Control ... dialog box. There is no password control unless there is a user defined as 'e3238s'.

See Also enableAccess (page 109)

downloadable

Syntax `downloadable: filename`

`The maximum length is 79 characters.`

Description The E9821A DSP module serves as the measurement engine for the E3238S system. As such, it manages the system tuner(s), the ADC, and its own DSP assets such that, given a set of sweep/search parameters provided by the host, it returns spectral data.

To perform this task, it requires an operating system and the measurement program that are downloaded from the host controller. This command specifies the DSP program file to be downloaded when the application starts.

Example The following command defines the downloadable file:

```
downloadable: C:\E3238s\downloadables\e9821a.out
```

e3238sService

Syntax `e3238sService: string`

`The maximum length is 79 characters.`

Description Specifies the name of the *service* used to define the port number and service provided. Socket server *services* are listed in a file: For Windows systems, the file is `\WinNT\system32\drivers\etc\Services`.

The line in the file may look like this:

```
e3238s          7011/tcp
```

The default value of this parameter is `e3238s`.

Example The following commands show an example socket configuration:

```
e3238sService:                e3238s
e3238sServiceMaxConnections: 4
e3238sServiceDataBufferSize: 512
e3238sServiceSendBufferSize: 0
e3238sServiceRecvBufferSize: 0
```

See Also
e3238sServiceDataBufferSize (page 105)
e3238sServiceMaxConnections (page 106)
e3238sServiceRecvBufferSize (page 107)
e3238sServiceSendBufferSize (page 108)
maxClientSockets (page 124)
maxServices (page 125)
socketServer (page 172)
socketServerTimerInterval (page 173)

e3238sServiceDataBufferSize

Syntax `e3238sServiceDataBufferSize: integer`

`512 ≤ integer ≤ 4194304`

Description Specifies the maximum size of the buffer used to hold incoming data (from the sockets receive buffer). For the E3238S, this data amounts to incoming commands.

The default value of this parameter is 512.

Example The following commands show an example socket configuration:

```
e3238sService:                e3238s
e3238sServiceMaxConnections: 4
e3238sServiceDataBufferSize: 512
e3238sServiceSendBufferSize: 0
e3238sServiceRecvBufferSize: 0
```

See Also
e3238sService (page 104)
e3238sServiceMaxConnections (page 106)
e3238sServiceRecvBufferSize (page 107)
e3238sServiceSendBufferSize (page 108)
maxClientSockets (page 124)
maxServices (page 125)
socketServer (page 172)
socketServerTimerInterval (page 173)

e3238sServiceMaxConnections

Syntax `e3238sServiceMaxConnections: integer`

`1 ≤ integer ≤ 10`

Description Specifies the maximum number of server sockets available on the E3238S host. The E3238S Sockets Connections dialog box shows the maximum number of connections and any clients connected to the E3238S service.

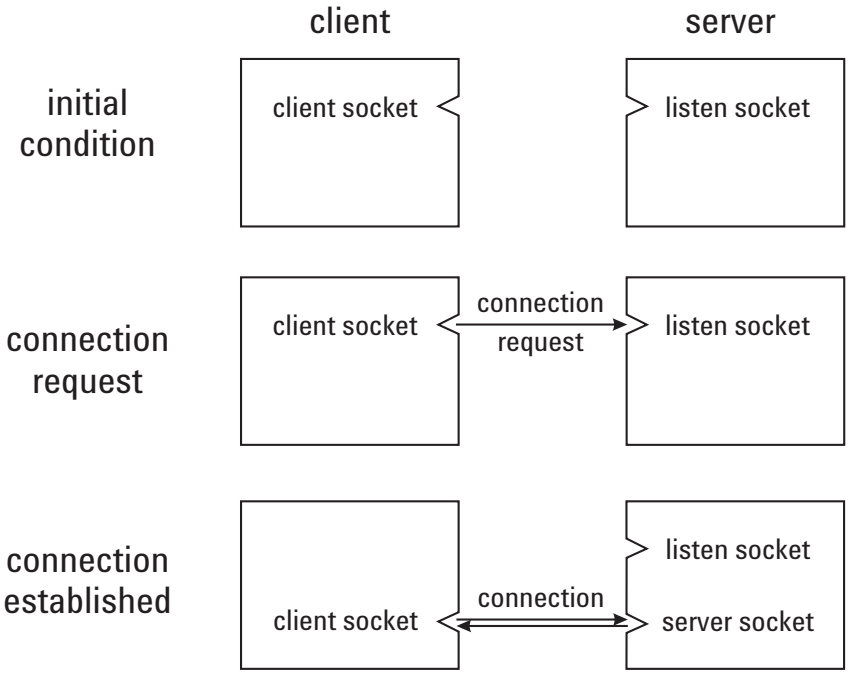
This parameter's default value is 4.

This setting has no impact on the socketServerClientSockets parameter. In fact, a more appropriate name would be socketServerMaxServerSockets.

Example The following commands show an example socket configuration:

```
e3238sService:                 e3238s
e3238sServiceMaxConnections: 4
e3238sServiceDataBufferSize: 512
e3238sServiceSendBufferSize: 0
e3238sServiceRecvBufferSize: 0
```

Figure 47. Socket connection process



See Also

e3238sService (page 104)
e3238sServiceDataBufferSize (page 105)
e3238sServiceRecvBufferSize (page 107)
e3238sServiceSendBufferSize (page 108)
maxClientSockets (page 124)
maxServices (page 125)
socketServer (page 172)
socketServerTimerInterval (page 173)

e3238sServiceRecvBufferSize

Syntax `e3238sServiceRecvBufferSize: integer`

`0 ≤ integer ≤ 8388608`

Description Specifies the number of bytes to allot for the purpose of receiving packets at the operating system level.

The default value for this parameter is 0 which allows the system to adjust the actual value used to match the conditions. The default value for Windows® is 8192.

Example The following commands show an example socket configuration:

```
e3238sService:                e3238s
e3238sServiceMaxConnections: 4
e3238sServiceDataBufferSize: 512
e3238sServiceSendBufferSize: 0
e3238sServiceRecvBufferSize: 0
```

See Also

e3238sService (page 104)
e3238sServiceDataBufferSize (page 105)
e3238sServiceMaxConnections (page 106)
e3238sServiceSendBufferSize (page 108)

maxClientSockets (page 124)
maxServices (page 125)
socketServer (page 172)
socketServerTimerInterval (page 173)

e3238sServiceSendBufferSize

Syntax `e3238sServiceSendBufferSize: integer`

`0 ≤ integer ≤ 8388608`

Description Specifies the number of bytes to allot to for the purpose of sending packets. This allows you to select a value to optimize performance given the data rate of the LAN. If the rate is low you may want to choose a large value for this parameter.

The default value for this parameter is 0 which allows the system to adjust the actual value used to match the conditions. The default value for Windows® is 8192.

Example The following commands show an example socket configuration:

```
e3238sService:                e3238s
e3238sServiceMaxConnections: 4
e3238sServiceDataBufferSize: 512
e3238sServiceSendBufferSize: 0
e3238sServiceRecvBufferSize: 0
```

See Also

e3238sService (page 104)
e3238sServiceDataBufferSize (page 105)
e3238sServiceMaxConnections (page 106)
e3238sServiceRecvBufferSize (page 107)

maxClientSockets (page 124)
maxServices (page 125)
socketServer (page 172)
socketServerTimerInterval (page 173)

enableAccess

Syntax `enableAccess: string`

`The maximum length is 255 characters.`

Description Used to restore access to features within the E3238S application that have been restricted through use of the disableAccess (page 102). You can also choose to just comment out the disableAccess lines in the e3238s.cfg file

See Also disableAccess (page 102)

energyHistoryFilter

Syntax

```
energyHistoryFilter: filename

The maximum length is 511 characters.
```

Description

Specifies path and filename(s) of shared library program(s) used to filter entries in the Energy History.

As many as 16 filter definitions may be loaded but no more than 5 of each type (pre or post) may be active at a time.

Notes

If a library contains code for both features and filters, use the same name with both commands.

Either regular slashes (/) or back slashes (\) may be used in the pathname.

Example

```
energyHistoryFilter: /e3238s/filterBUTCH.dll, \
                      /e3238s/filterSUNDANCE.dll

featureExtraction:  /e3238s/featureBUTCH.dll, \
                    /e3238s/featureSUNDANCE.dll
```

See Also

featureExtraction (page 111)

featureExtraction

Syntax `featureExtraction: filename`

`The maximum length is 511 characters.`

Description Specifies path and filename(s) of shared library program(s) used to extract features from raw spectral search data.

Example

```
featureExtraction:  /e3238s/featureBUTCH.dll, \
                    /e3238s/featureSUNDANCE.dll
```

As many as 4 feature shared libraries may be loaded, each of which may define as many as 4 features.

Note If one shared-library program contains code for both features and filters, use the same name with both commands.

Note You may use either regular slashes (/) or back slashes (\) in the pathname.

See Also energyHistoryFilter (page 110)

genericLib

Syntax `genericLib: filename`

`The maximum length is 511 characters.`

Description Specifies a shared library comprising features that do not belong in any of the specific user-defined categories such as panes, feature types, feature filters, alarm tasks, and signal processing.

As many as four generic shared libraries may be loaded.

Example `genericLib: C:/e3238s/examples/doItAll.dll`

handoffRx.driver

Syntax `handoffRx(1..16).driver: filename`

`The maximum length is 79 characters.`

Description Specifies the shared-library file containing the driver code for a specific handoff receiver.

As many as 100 handoff receivers may be controlled by the E3238S system. The number 16 that appears in the syntax above is determined by the maxHandoffRxs parameter setting in the `E3238s` resource file. An error occurs when you specify more than this setting.

Example The following commands define the hardware configuration for a handoff receiver.

```
handoffRx1.driver:    C:/e3238s/lib/HD_8607.dll
handoffRx1.interface: rs232,COM1,9600
handoffRx1.label:     VHF/UHF  Rx
```

See Also handoffRx.interface (page 114)
handoffRx.label (page 115)

handoffRx.interface

Syntax `handoffRx(1..16).interface: string`

`The maximum length is 63 characters.`

Description Specifies the interface type connecting the handoff receiver to the E3238s system. Some examples are hpib, vxi, and com1.

As many as 100 handoff receivers may be controlled by the E3238S system. The number 16 that appears in the syntax above is determined by the maxHandoffRxs parameter setting in the `E3238s` resource file. An error occurs when you specify more than this setting.

Example The following commands define the hardware configuration and driver program for a handoff receiver:

```
handoffRx1.driver:    C:/e3238s/lib/HD_8607.dll
handoffRx1.interface: rs232,COM1,9600
handoffRx1.label:     VHF/UHF  Rx
```

Note When using a controller with a firewire interface, use “ASRL1” instead of “COM1” in the interface parameter. (The firewire IO interface does not support an alias for the serial port name.)

See Also handoffRx.driver (page 113)
handoffRx.label (page 115)

handoffRx.label

Syntax `handoffRx(1..16).label: string`

`The maximum length is 31 characters.`

Description Specifies a label for the handoff receiver listing in the handoff receiver pane.

As many as 100 handoff receivers may be controlled by the E3238S system. The number 16 that appears in the syntax above is determined by the maxHandoffRxs parameter setting in the `E3238s` resource file. An error occurs when you specify more than this setting.

Example

```
handoffRx1.driver:    C:/e3238s/lib/HD_8607.dll
handoffRx1.interface: rs232,COM1,9600
handoffRx1.label:     VHF/UHF  Rx
```

See Also handoffRx.label (page 115)
handoffRx.interface (page 114)

irigCableDelay

Syntax `irigCableDelay: integer`

`-9999999 ≤ integer ≤ 9999999`

Description Specifies the amount of time to allow for an IRIG time reference signal due to cable length between the IRIG source and this system.

Units are integers where each increment is equivalent to 100 ns (10e-7 s).

Note This selection has no effect when timeReference (page 175) is *systemClock*.

Example

```
timeReference:                  vxiIRIG
irigModel:                      bc350vxi
irigInterfaceParm:              200
irigOperatingMode:              Decode
irigTimeCodeFormat:             IRIGB
irigTimeCodeModulation:         AM
irigClock:                      Internal
irigSecondsFromGMT:             0
irigCableDelay:                 0
```

irigClock

Syntax `irigClock: enum`

`enum = {Internal, External}`

Description Specifies the source of the clock signal for the IRIG time reference module. This signal is used when irigOperatingMode (page 120) is *Freerun* or when decoding a reference signal that ceases.

- **Internal** means the onboard crystal oscillator is used as the time clock. This oscillator may be stabilized with a crystal oven.
- **External** means the signal on the front panel 10 MHz SMB connector is used as the time clock.

Note This selection has no effect when timeReference (page 175) is *systemClock*.

Example

```
timeReference:                  vxiIRIG
irigModel:                      bc350vxi
irigInterfaceParm:              200
irigOperatingMode:              Decode
irigTimeCodeFormat:             IRIGB
irigTimeCodeModulation:         AM
irigClock:                      Internal
irigSecondsFromGMT:             0
irigCableDelay:                 0
```

irigInterfaceParm

Syntax `irigInterfaceParm: integer`

`1 ≤ integer ≤ 254`

Description Specifies the logical address for the IRIG VXI module.

See the Hardware Installation section for information on configuring the module switches.

Note This selection is ignored when timeReference (page 175) is *systemClock*.

Example

```
timeReference:                vxiIRIG
irigModel:                    bc350vxi
irigInterfaceParm:            200
irigOperatingMode:            Decode
irigTimeCodeFormat:           IRIGB
irigTimeCodeModulation:       AM
irigClock:                    Internal
irigSecondsFromGMT:           0
irigCableDelay:               0
```

irigModel

Syntax `irigModel: enum`

`enum = {None, BC350VXI}`

Description Specifies the model number of the IRIG (time reference) VXI module.

The bc350VXI Time and Frequency Processor (TFP) by Datum Inc is the only IRIG time reference module supported.

Note This selection has no effect when timeReference (page 175) is *systemClock*.

Example

```
timeReference:                vxiIRIG
irigModel:                    bc350vxi
irigInterfaceParm:            200
irigOperatingMode:            Decode
irigTimeCodeFormat:           IRIGB
irigTimeCodeModulation:       AM
irigClock:                    Internal
irigSecondsFromGMT:           0
irigCableDelay:               0
```

irigOperatingMode

Syntax `irigOperatingMode: enum`

`enum = {Decode, Freerun, 1PPS, RealtimeClock, DigitalSync, GPSOnboard, GPSinAntenna}`

Description Specifies the method used by the time and frequency processor (TFP) to determine the time reference.

- **Decode** means the TFP decodes the IRIG timecode signal on its front panel connector to generate time values. The TFP locks its crystal oscillator to the input code rate. If the input code becomes unavailable, the TFP continues using the "flywheel" method running at the last known code rate. See Datum documentation for more information.

  Decode type is specified with the irigTimeCodeFormat (page 122) command.
- **Freerun** means the TFP generates time values using the clock specified with the irigClock (page 117) command.
- **1PPS** means the TFP synchronizes to the signal on the 1PPS front panel input (pin 14 of J1).
- **RealtimeClock** means the TFP synchronizes to the onboard real time clock (RTC) that is battery-backed. This mode is not recommended when a crystal oven is installed because the RTC accuracy is less than that of the oscillator.
- **GPSOnboard** is not currently supported.
- **GPSinAntenna** means the TFP uses a global positioning system signal to generate time values (receiver at antenna). The GPS is available separately.

Note This selection is ignored when timeReference (page 175) is *systemClock*.

Example

```
timeReference:                vxiIRIG
irigModel:                    bc350vxi
irigInterfaceParm:            200
irigOperatingMode:            Decode
irigTimeCodeFormat:           IRIGB
irigTimeCodeModulation:       AM
irigClock:                    Internal
irigSecondsFromGMT:           0
irigCableDelay:               0
```

irigSecondsFromGMT

Syntax `irigSecondsFromGMT: integer`

`-86399 ≤ integer ≤ 86399`

Description Specifies the number of seconds by which to shift the time reference to get GMT.

All time stamp values are given in GMT. When the system time reference signal is a local time, this parameter may be used to derive GMT such that

`time_ref + irigSecondsFromGMT = GMT`

Example

```
timeReference:                  vxiIRIG
irigModel:                      bc350vxi
irigInterfaceParm:              200
irigOperatingMode:              Decode
irigTimeCodeFormat:             IRIGB
irigTimeCodeModulation:         AM
irigClock:                      Internal
irigSecondsFromGMT:             0
irigCableDelay:                 0
```

irigTimeCodeFormat

Syntax `irigTimeCodeFormat: enum`

`enum = {IRIGA, IRIGB, 2137, NASA36, XR3}`

Description Specifies the time reference signal code format applied to the time code input (pin 7 of J1).

- **IRIGA** specifies IRIG Standard Format A 1000 PPS code. Reference IRIG document 200-95.
- **IRIGB** specifies IRIG Standard Format B 100 PPS code. Reference IRIG document 200-95.
- **NASA36** specifies NASA 36-bit one-second time code.
- **2137** specifies 25 PPS one-second time code (1 kHz). (Same as XR3 with 100 Hz symbol rate.)
- **XR3** specifies 25 PPS one-second time code (250 Hz). (100 Hz symbol rate.)

Note This selection is relevant only when irigOperatingMode (page 120) is *Decode*.

Example

```
timeReference:                vxiIRIG
irigModel:                    bc350vxi
irigInterfaceParm:            200
irigOperatingMode:            Decode
irigTimeCodeFormat:           IRIGB
irigTimeCodeModulation:       AM
irigClock:                    Internal
irigSecondsFromGMT:           0
irigCableDelay:               0
```

irigTimeCodeModulation

Syntax `irigTimeCodeModulation: enum`

`enum = {AM, PCM}`

Description Specifies type of time code modulation used on the time reference signal.

- **AM** specifies amplitude modulation.
- **PCM** specifies pulse code modulation or DC level shift (DCLS). This modulation type is not supported when irigTimeCodeFormat (page 122) is *2137* or *XR3*.

Note This selection is ignored when timeReference (page 175) is *systemClock*.

Example

```
timeReference:                  vxiIRIG
irigModel:                      bc350vxi
irigInterfaceParm:              200
irigOperatingMode:              Decode
irigTimeCodeFormat:             IRIGB
irigTimeCodeModulation:         AM
irigClock:                      Internal
irigSecondsFromGMT:             0
irigCableDelay:                 0
```

maxClientSockets

Syntax

```
maxClientSockets: integer

0 ≤ integer ≤ 16
```

Description

Specifies the maximum number of client sockets that can be open in the E3238S application. The default is 0. However, a user-defined library may use client sockets to connect to external services (typically on other systems).

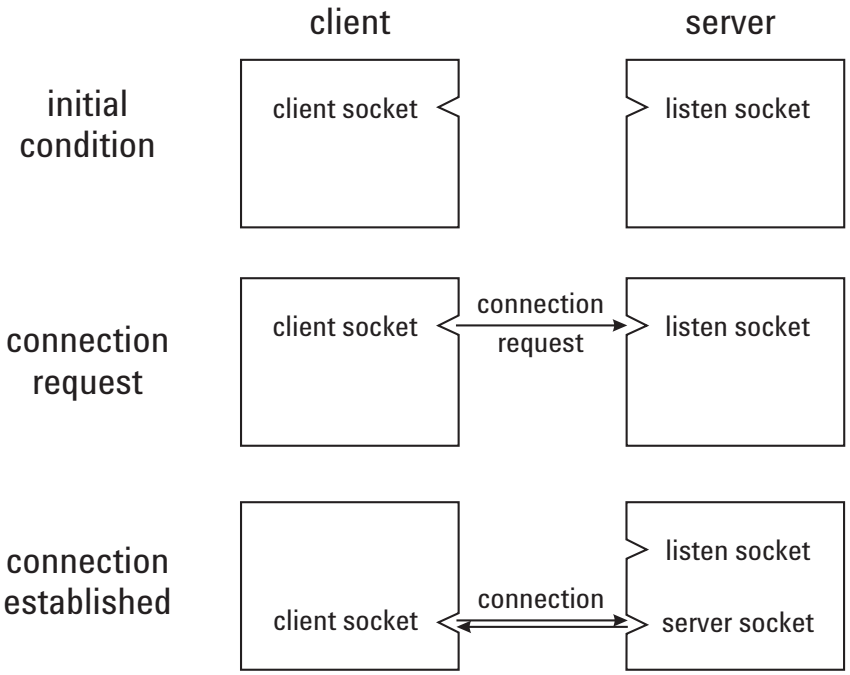
Figure 48 illustrates the socket startup process and the various socket types. The number of 'listen sockets' corresponds to the maxServices (page 125) setting.

This setting has no impact on the e3238sServiceMaxConnections parameter described on page 106.

Example

```
socketServer:                Disabled
maxServices:                 1
maxClientSockets:            0
socketServerTimerInterval:   5
```

Figure 48. Socket connection process



See Also

maxServices (page 125)
socketServer (page 172)
socketServerTimerInterval (page 173)
e3238sService (page 104)
e3238sServiceDataBufferSize (page 105)
e3238sServiceMaxConnections (page 106)
e3238sServiceRecvBufferSize (page 107)
e3238sServiceSendBufferSize (page 108)

maxServices

Syntax

`maxServices: integer`

`1 ≤ integer ≤ 5`

Description

Specifies the number of sockets used to listen for connections.

The default value of this parameter is 1.

Only one listen socket is required for 1-10 clients to the E3238S service.

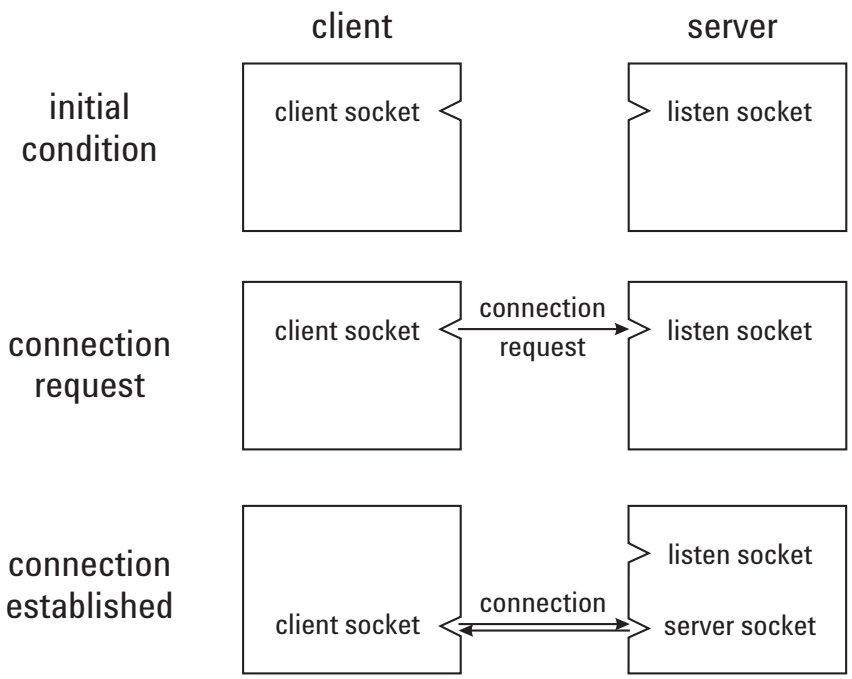
There should be 1 listen socket for every service provided by the system (see `\WINNT\system32\drivers\etc\services`).

This number may be incremented to support additional socket services implemented through user-defined shared-library programs.

Example

```
socketServer:                Disabled
maxServices:                 1
maxClientSockets:            0
socketServerTimerInterval:   5
```

Figure 49. Socket connection process



See Also

maxClientSockets (page 124)
socketServer (page 172)
socketServerTimerInterval (page 173)
e3238sService (page 104)
e3238sServiceDataBufferSize (page 105)
e3238sServiceMaxConnections (page 106)
e3238sServiceRecvBufferSize (page 107)
e3238sServiceSendBufferSize (page 108)

modRec.alias (EMR)

Syntax

```
modRec(1..32).alias: string
The maximum length is 31 characters.
```

Description

Specifies an alternate name to be used in the application's user interface so that the real signal name is not displayed.

Example

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                      Modulation Recognition                                  !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

modRec1.enabled:        False
modRec1.hostLib:        C:\E3238s\mr1\mr1.dll
modRec1.args:
modRec1.alias:           Bob
```

Note

You may use either regular slashes (/) or back slashes (\) in the pathname.

See Also

modRec.args (EMR) (page 127)
modRec.enabled (EMR) (page 128)

modRec.args (EMR)

Syntax `modRec(1..32).args: string`

`The maximum length is 255 characters.`

Description Specifies the values for any parameters that a custom library might pass in.

Example

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                        Modulation Recognition                               !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

modRec1.enabled:       False
modRec1.hostLib:       C:\E3238s\mr1\mr1.dll
modRec1.args:
modRec1.alias:           Bob
```

Note You may use either regular slashes (/) or back slashes (\) in the pathname.

See Also modRec.alias (EMR) (page 126)
modRec.enabled (EMR) (page 128)

modRec.enabled (EMR)

Syntax

```
modRec(1..32).enabled: enum
enum = {False, True}
```

Description

Specifies whether the signal processing components are to be loaded. This allows you to disable a signal library without commenting out all the lines associated with it.

Another way to disable a signal library is to leave the setting True and just comment out the signal.enabled line.

Example

The following example shows a signal that is disabled:

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                      Modulation Recognition                                 !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

modRec1.enabled:        False
modRec1.hostLib:        C:\E3238s\mr1\mr1.dll
modRec1.args:
modRec1.alias:            Bob
```

Note

You may use either regular slashes (/) or back slashes (\) in the pathname.

See Also

modRec.alias (EMR) (page 126)
modRec.args (EMR) (page 127)

modRec.hostLib (EMR)

Syntax

```
modRec(1..32).hostLib: filename
The maximum length is 127 characters.
```

Description Specifies the filename of the Mod Rec library to be loaded in the host (system controller). This provides the host component of the typical host-target interaction mechanism.

Example

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                     Modulation Recognition                                   !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

modRec1.enabled:       False
modRec1.hostLib:       C:\E3238s\mr1\mr1.dll
modRec1.args:
modRec1.alias:           Bob
```

Note You may use either regular slashes (/) or back slashes (\) in the pathname.

multiSystemSyncTimeout (EMS)

Syntax `multiSystemSyncTimeout: integer`

`1 ≤ integer ≤ 60`

Description Specifies the time (in seconds) to wait for system synchronization acknowledgment in a multi system configuration.

Example `multiSystemSyncTimeout: 5`

multiSystemSynchronization (EMS)

Syntax `multiSystemSynchronization: enum`
`enum = {Off, Master, Slave}`

Description Specifies whether the system is either a master or slave system

Example
```
multiSystemSynchronization: Master
-or-
multiSystemSynchronization: Slave
```

multipleSignalsPerProcessor

Syntax `multipleSignalsPerProcessor: enum`

`enum = {Disabled, Enabled}`

Description Specifies whether you can process more than one signal type on a G4 processor is activated.

Example

```
multipleSignalsPerProcessor: Enabled

asxDsp_0:   C:\E3238S\downloadables\ASXdsp_0.elf
asxDsp_1:   C:\E3238S\downloadables\ASXdsp_1.elf

searchRx1.minDelayTimeRequired: 0

!
! Demo Signal Type
!

signal1.enabled:     False
signal1.hostLib:     C:\E3238S\demo\demoHost.dll
signal1.hostDsp:     C:\E3238S\demo\demoDsp.dll
signal1.targetDsp:   C:\E3238S\demo\demoDsp.esl
signal1.loadFactor:  64
signal1.minChannels: 4
signal1.maxChannels: 32
signal1.args:
```

See Also signal.targetDsp (page 171)

asxDsp_0 (page 92)

numSearchRx (EMC)

Syntax `numSearchRx: integer`

`1 ≤ integer ≤ 4`

Description Specifies the number of search channels installed.

Tuner, ADC, and DSP modules should be specified for each channel.

Example

```
searchRxConfiguration: MultiChannelParallelTimeSynchronousSearch
!
numSearchRx:2

searchRx1.adcModel:             E1439B/70
searchRx1.adcInterfaceParm:     130
searchRx1.adcClock:             Internal
searchRx1.adcDataPort:          localBus
searchRx1.adcMasterClock:       Auto
!
searchRx2.adcModel:             E1439B/70
searchRx2.adcInterfaceParm:     131
searchRx2.adcClock:             Internal
searchRx2.adcDataPort:          localBus
searchRx2.adcMasterClock:       Auto
```

searchRx.adcClock

Syntax `searchRx1.adcClock: enum`

`enum = {Internal, External}`

Description Specifies the clock source for the analog-to-digital converter in the search receiver.

- **Internal** means the ADC will use it's own internal reference. This is the default selection.
- **External** means the ADC will use the signal present on its Ext Clock connector as a clock for the sample rate. This signal may be generated by a source in the system (e.g., a downconverter's LO Ref) or may come from a source outside the system.
  - E1437A: 10.24 MHz
  - E1438A/B, E1439A/B/D: 10 MHz
  - N6830A: 10.0 MHz (Master) or 5.0 MHz (Slave)

The WJ9119 HF LO module provides an accurate clock signal from its ADC CLK outputs which can be used as a reference for the E1437A ADC.

Example

```
searchRx1.adcModel:             E1437A
searchRx1.adcInterfaceParm:     130
searchRx1.adcClock:             Internal
searchRx1.adcDataPort:          localBus
searchRx1.adcMasterClock:   Auto
```

searchRx.adcDataPort

Syntax

```
searchRx1.adcDataPort: enum

enum = {VXI, LocalBus, FPDP}
```

Description

Specifies the data path between the ADC and DSP modules.

- **VXI** specifies using the bi-directional VXI bus on the VME backplane. This bus is also used to send commands from the controller to the VXI modules and to move the DSP output data back to the host/controller.

  VXI is **not** the preferred path for the ADC data to use (it's a bottleneck for the DSP output and the ADC data can be *much* larger) but it can be used when the others are not available; e.g. local bus hardware failure.

- **LocalBus** is a VXI backplane bus optimized for data throughput. Data flows between slots, from left to right, only. The modules must also be installed side-by-side since data cannot flow through an empty slot.

- **FPDP** (front panel data port) is a serial, fiber-optic interface that may be used when both the ADC and the DSP modules support it. The FPDP throughput is much greater than that of the Local Bus.

The FPDP interface is supported on the E1438B/D, E1439B/D, and the N6830A ADCs and the E9821A DSP.. When the FPDP port is used, both the receive (Rx) and transmit (Tx) fiber-optic lines must be connected between the ADC and the DSP (Tx-to Rx). The ADC sends data to the DSP and the DSP sends flow control information to the ADC.

Note

This command defines *both* the ADC data output port *and* the DSP input ports.

Example

The following commands define the hardware configuration for the ADC in the search receiver.

```
searchRx1.adcModel:             E1439B/70
searchRx1.adcInterfaceParm:     133
searchRx1.adcClock:             Internal
searchRx1.adcDataPort:          FPDP
searchRx1.adcMasterClock:       Auto
```

See Also

searchRx.dspDataPort (page 145)

searchRx.dspCmndPort (page 143)

searchRx.adcInterfaceParm

Syntax `searchRx1.adcInterfaceParm: integer`

`1 ≤ integer ≤ 254`

Description Specifies the VXI logical address used to communicate with the ADC module in the search receiver. This address must correspond with switches set inside the module.

Example The following commands define the hardware configuration for the ADC in the search receiver.

```
searchRx1.adcModel:            E1437A
searchRx1.adcInterfaceParm:    130
searchRx1.adcClock:          Internal
```

See Also Tuner Configuration (page 45)

searchRx.adcMasterClock

Syntax `searchRx1.adcMasterClock: enum`

`enum = {Off, On, Auto}`

Description Specifies whether the search ADC provides its sample clock for use by other sampling modules such as other ADCs (in a multi-channel system) or the DDC (digital downconverter) to use.

Whether the clock is provided on the ADC front panel (SMB connectors) or the VXI backplane is specified with the

Note Only one ADC may drive the VXI backplane clock line at a time.

- **Off** specifies that the search ADC's clock signal is **not** put on the VXI backplane.
- **On** specifies that the search ADC's clock signal **is** put on the VXI backplane.
- **Auto** specifies that the search ADC's clock is put on the VXI backplane. The clock is used by the DDCs in the channelizer system.

Example The following commands define the hardware configuration for the ADC in the search receiver.

```
searchRx1.adcModel:            E1437A
searchRx1.adcInterfaceParm:    130
searchRx1.adcClock:            Internal
searchRx1.adcDataPort:         localBus
searchRx1.adcMasterClock:         Auto
```

searchRx.adcModel

Syntax

```
searchRx1.adcModel: enum

enum = {E1437A, E1438A, E1438B, E1439A/70, E1439B/70, E1439C/70,
E1439D/70, E1439A/BB, E1439B/BB, E1439C/BB, E1439D/BB, N6830A/70,
N6830A/HF}
```

Description

Specifies the model number of the analog-to-digital converter module in the search receiver.

As of this printing, the ADC modules supported are as follows:

- E1437A: f_s = 20.48 MHz, span = 8 MHz
- E1438A: f_s = 102.4 MHz, span = 40 MHz
- E1438B: f_s = 102.4 MHz, span = 40 MHz
- E1438D: f_s = 102.4 MHz, span = 40 MHz
- E1439A/70: f_s = 95 MHz, span = 36 MHz, f_c = 70 MHz
- E1439B/70: f_s = 95 MHz, span = 36 MHz, f_c = 70 MHz
- E1439D/70: f_s = 95 MHz, span = 36 MHz, f_c = 70 MHz
- E1439A/BB: f_s = 95 MHz, span = 36 MHz, BB = baseband)
- E1439B/BB: f_s = 95 MHz, span = 36 MHz, BB = baseband)
- E1439D/BB: f_s = 95 MHz, span = 36 MHz, BB = baseband)
- N6830A/HF: f_s = 81.92 MHz, span = 32 MHz
- N6830A/70: f_s = 95 MHz, span = 36 MHz

B models have the fiber-optic, front panel data port (FPDP).
C models have no Local Bus or FPDP interfaces and are not recommended.
D models are the same as the B model with improved phase noise performance.

The E1439 has a 70 MHz IF input but may also be used in baseband mode. The IF input allows it to be used with the E2730A, E2731A, or CS5040 tuners. The baseband mode bypasses the IF section of the input including the attenuators. See the following note.

Example

The following commands define the hardware configuration for the ADC in the search receiver.

```
searchRx1.adcModel:             E1437A
searchRx1.adcInterfaceParm:     130
searchRx1.adcClock:             Internal
searchRx1.adcDataPort:          localBus
searchRx1.adcMasterClock:       Auto
```

Note

When the E1439 is used in baseband, the ADC's fullscale input is -21 dBm and there is no input attenuation so be sure to limit the input signal to avoid overloading the ADC.

searchRx.adcSampleRate

Syntax

```
searchRx1.adcSampleRate: integer

[10240000, 20480000, 40960000, 81920000, etc (see below)]
```

Description

Specifies the ADC sample rate for the N6830A. This parameter will affect the bandwidth available for narrowband signal processing.

For best probability of intercept (fastest search revist times) use the lowest stare bandwidth that covers the frequency range of interest.

N6830A/HF

Sample Rate	Stare Bandwidth
81920000	32 MHz
40960000	16 MHz
20480000	8 MHz
10240000	4 MHz

N6830A/70

Sample Rate	Stare Bandwidth
95000000	36 MHz
47500000	18 MHz
23750000	9 MHz
11875000	4.5 MHz

If you are upgrading from a 9119-1 tuner with E1437 ADC to an N6830A/HF, use the following settings.

```
searchRx1.adcModel:      N6830A/HF
searchRx1.adcSampleRate: 20480000
searchRx1.adcDataPort:   FPDP
```

Increasing the ADC sample rate may affect the sweep rate depending on the search setup number of averages and RBW selections.

Example

```
searchRx1.adcModel:          N6830A/HF
searchRx1.adcInterfaceParm:  130
searchRx1.adcSampleRate:     20480000
searchRx1.adcDataPort:       FPDP
searchRx1.adcMasterClock: Auto
```

searchRx.antenna.calFile

Syntax `searchRx1.antenna(1..16).calFile: filename`

`The maximum length is 79 characters.`

Description Specifies the filename of the calibration data to be used for a given antenna.

Example

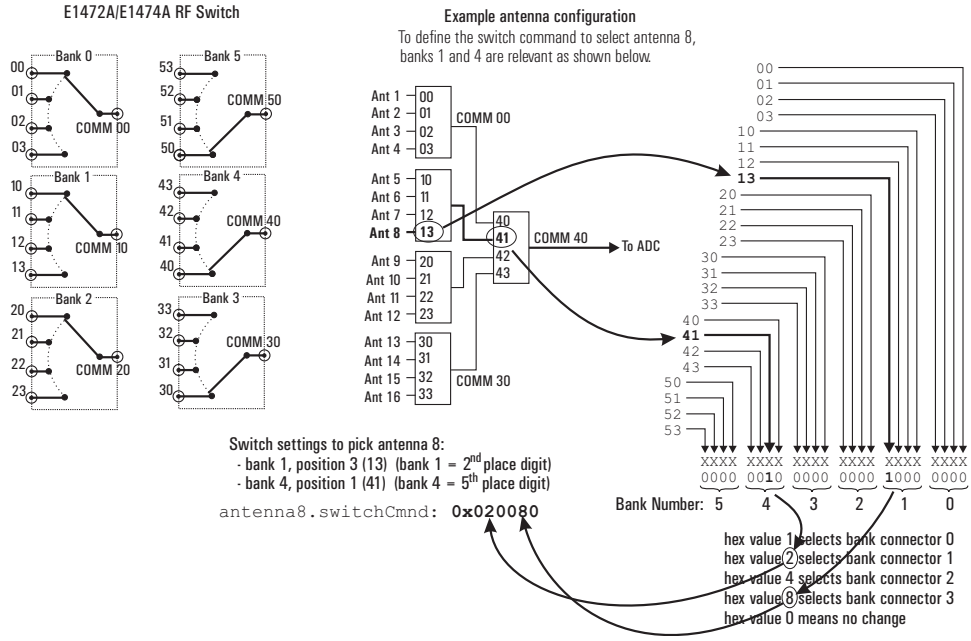
```
!These commands control the application of corrections to compensate for
!tuner response and antenna path response. These user-supplied
!corrections are in addition to built-in RF and IF corrections. For an
!example of the file format, see the file e3238s\cal\d.tuner1.cal.

!Cal files are normally located in the C:\E3238s\cal

searchRx1.antenna1.calFile:    antenna1.cal
```

searchRx.antenna.switchCmnd

Syntax `searchRx1.antenna(1..16).switchCmnd: integer`

Description This command is used to program the RF switch. The value is logically compared with searchRx.antenna.switchMask (page 142) to determine which switches must be changed such that only those are changed when the antenna selection changes. This approach is used to minimize relay activity and maximize switch contact life. The parameter value may be given either as an integer or hexadecimal. The hexadecimal value should be preceded with "0x" as shown in the figure and example.

Figure 50. Antenna switch command translation



This hexadecimal number uses one digit to define a connection for each of the six banks in the switch. Valid values for a single hex digit are 0, 1, 2, 4, or 8 (0 = don't change). See figure 50.

Example The hex value to specify using connection 13, and 41 is 0x020080:
- bank 1 is 2nd place from right, 8 here selects connection 13
- bank 4 is 5th place from right, 2 here selects connection 41

The integer (decimal) equivalent value of 0x020080 is 131,200.

Each antenna must have a switchCmnd and switchMask as shown for antennas 1, 2, and 8 in the following example:

```
searchRx1.antenna1.switchCmnd:          0x00000000
searchRx1.antenna1.switchMask:          0x0000F00F
searchRx1.antenna2.switchCmnd:          0x00000002
searchRx1.antenna2.switchMask:          0x0000F00F
searchRx1.antenna8.switchCmnd:          0x00002080
searchRx1.antenna8.switchMask:             0x0000F0F0
```

searchRx.antenna.switchMask

Syntax `searchRx1.antenna(1..16).switchMask: integer`

Description This command is used to program the RF multiplex switch, which allows antenna switching. It is used in conjunction with searchRx.antenna.switchCmnd (page 141) to minimize relay activity and maximize switch contact life.

Example

```
searchRx1.antenna1.switchCmnd:          0x00010001
searchRx1.antenna1.switchMask:          0x000F000F
searchRx1.antenna2.switchCmnd:          0x00010002
searchRx1.antenna2.switchMask:             0x000F000F
```

searchRx.dspCmndPort

Syntax

```
searchRx1.dspCmndPort: string
```

The maximum length is 127 characters.

Description

Specifies the path through which the DSP module gets commands from the host (system controller). The acceptable strings are:

- VXI

Other options will be available in future releases of the E3238 application.

VXI specifies using the bi-directional VXI bus on the VME backplane. This has been the default command port for VXI modules up to now. This bus may also be used to move the DSP output data to the host/controller.

Example

The following commands define the hardware configuration for the search receiver DSP module:

```
searchRx1.dspModel:             E9821A
searchRx1.dspModules:           128
searchRx1.dspNoHardwareConfig: 222D
searchRx1.dspCmndPort:          VXI
searchRx1.dspDataPort:          VXI
searchRx1.dspDataCompression:   On
downloadable:                   C:\E3238s\downloadables\e9821a.out
```

See Also

searchRx.dspDataPort (page 145)

searchRx.adcDataPort (page 135)

searchRx.dspDataCompression

Syntax

`searchRx(1..4).dspDataCompression: enum`

`enum = {Off, On}`

Description

This provides a compression feature that moves the data from the DSP module back to the controller much more efficiently.

The default setting is Off.

Data compression improves the search system performance when the limiting factor is the IO bandwidth between the DSP and the controller.

It also helps when there is a channelizer subsystem that passes data back to the controller via the VXI backplane because it frees up bandwidth.

The circumstances that benefit from data compression:

- High resolution settings (low RBW values) when DSP is not the limiting factor
- Multiple search channels (supported by option EMC)
- When used with a collection subsystem having many narrow-band channels. (Or you want to add more channels to an existing system that is already IO bound.)

The disadvantages:

- When performance is limited by the DSP processing resources (such as with a large number of averages and a high resolution setting) the compression algorithm may degrade search performance somewhat.
- The compression algorithm also impacts amplitude accuracy by as much as ±0.1 dBm.

Note

It is possible that an ASD feature extraction library may request time data. Time data is passed from the DSP to the controller via the same path as the magnitude data (the VXI backplane). Time data is not compressed when `dspDataCompression` is `On`.

searchRx.dspDataPort

Syntax

```
searchRx1.dspDataPort: string

The maximum length is 127 characters.
```

Description

Specifies the path through which the DSP passes data to the host (system controller). The acceptable strings are:

- `VXI`

Other options will be available in future releases of the E3238 application.

VXI specifies using the bi-directional VXI bus on the VME backplane. This has been the default command port for VXI modules up to now. This bus may also be used to move the DSP output data to the host/controller.

Example

The following commands define the hardware configuration for the search receiver DSP module:

```
searchRx1.dspModel:             E9821A
searchRx1.dspModules:           128
searchRx1.dspNoHardwareConfig: 222D
searchRx1.dspCmndPort:          VXI
searchRx1.dspDataPort:          VXI
searchRx1.dspDataCompression:   On
downloadable:                   C:\E3238s\downloadables\e9821a.out
```

See Also

searchRx.dspCmndPort (page 143)

searchRx.dspModel

Syntax `searchRx1.dspModel: enum`

`enum = {E9821A}`

Description Specifies the model of the digital signal processor module. The only model supported is the E9821A.

Example The following commands define the hardware configuration for the search receiver DSP module:

```
searchRx1.dspModel:              E9821A
searchRx1.dspModules:            128, 129
searchRx1.dspNoHardwareConfig: 222D
searchRx1.dspCmndPort:           VXI
searchRx1.dspDataPort:           VXI
searchRx1.dspDataCompression:    On
downloadable:                    C:\E3238s\downloadables\e9821a.out
```

Note For multi-channel systems that have multiple ADC's sharing one E9821A DSP, searchRx2.dspModel:, searchRx3.dspModel:, and so on should be set to **Shared**.

See Also The configuration of the E9821A is discussed under DSP Configuration (page 36).

"Multi-Channel Hardware Installation" on page 193

searchRx.dspModules

Syntax

```
searchRx(1..4).dspModules: string
The maximum length is 255 characters.
```

Description

Specifies the logical address(es) of the installed DSP module(s) as a comma-separated string of integers between 1 and 254.

Each address must correspond with switches set inside the modules. See E9821A VXI Signal Processing Module (page 42) for information on setting the module switches.

This defines all the DSP modules in the system, even though they may perform narrow-band, signal processing that is not, technically, part of the search receiver.

Example

```
searchRx1.dspModel:              E9821A
searchRx1.dspModules:            128,129
searchRx1.dspNoHardwareConfig: 222D
searchRx1.dspCmndPort:           VXI
searchRx1.dspDataPort:           VXI
searchRx1.dspDataCompression:    On
downloadable:                    C:\E3238s\downloadables\e9821a.out
```

See Also

The DSP module runs the program specified by the downloadable (page 103) command. This program controls most of the other VXI modules in the system.

searchRx.dspNoHardwareConfig

Syntax `searchRx(1..4).dspNoHardwareConfig: string`

`The maximum length is 79 characters.`

Description Specifies the type and location of PMC cards on the E9821 DSP modules when the E3238s application is run in the -noHardwre mode.

- E = empty
- 1 = single-G4 processor (option 110)
- 2 = dual-G4 processor (option 100)
- D = DDC (digital downconverter, option 200)

Example

```
searchRx1.dspModules:  128,129
searchRx1.dspNoHardwareConfig: 222D E2DE
```

This example illustrates two DSP modules configured as follows:

at logical address 128:

- site 1: dual-G4 processor
- site 2: dual-G4 processor
- site 3: dual-G4 processor
- site 4: 32-channel DDC

at logical address 129:

- site 1: empty
- site 2: dual-G4 processor
- site 3: 32-channel DDC
- site 4: empty

searchRx.minDelayTimeRequired

Syntax

```
searchRx(1..4).minDelayTimeRequired: float
0 ≤ float ≤ 3600
```

Description

Specifies minimum amount of delay (in seconds) required for signal processing.

When one or more narrowband signal processing libraries are loaded, the application allocates narrowband assets based on the number of libraries loaded and the various channel requirements associated with each, as well as the number of DDC channels and G4 processors that are installed.

The amount of delay required is typically determined by calculating the time needed to detect the energy and verify that the signal is the correct type.

Delay is provided by buffering the wideband data in memory on a processor card “in front of” a DDC card. See figure 25 on page 36. The minimum delay set in the default config file is 0 seconds, and disabled.

When set to 0, the Delay tab in Search Reciever Configuration will be ghosted, or disabled.

To get delay to work also requires specifying parameters in the

Example

The following example shows a minimum delay of .1 seconds.

```
searchRx1.minDelayTimeRequired: .1
```

searchRx.snapshotMemory.interfaceParm

Syntax `searchRx1.snapshotMemory(1..10).interfaceParm: integer`

`1 ≤ integer ≤ 254`

Description Specifies the VXI logical address used to communicate with the snapshot memory module.

See the Hardware Installation section for information on how to set these switches.

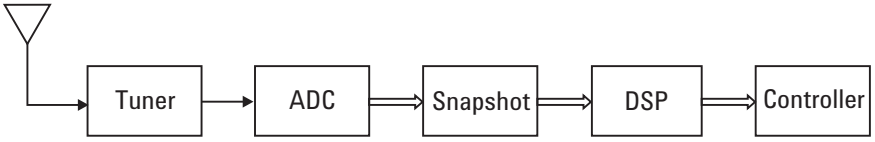
Note The E9830A is obsolete. This configuration parameter is provided to support the continued use of previously-purchased units.

Example The following command defines a snapshot module in the `e3238s.cfg` file:

```
searchRx1.snapshotMemory1.model:          E9830A
searchRx1.snapshotMemory1.interfaceParm: 133
```

The example above defines the configuration of one E9830A (Snapshot) module shown in figure 51. Because it's input is the VXI Local Bus, it must be installed between the ADC and the DSP modules in the VXI chassis.

Figure 51. Snapshot memory block diagram



Note The E9830A module does not have the fiber-optic FPDP

See Also searchRx.snapshotMemory.model (page 151)

searchRx.snapshotMemory.model

Syntax

```
searchRx1.snapshotMemory(1..10).model: enum

enum = {None, E9830A}
```

Description

Specifies the model of the snapshot memory module(s) installed. A snapshot module is used to collect/capture time data. As many as ten modules may be installed.

The only valid entry for this parameter is E9830A.

Note

The E9830A is obsolete. This configuration parameter is provided to support the continued use of previously-purchased units.

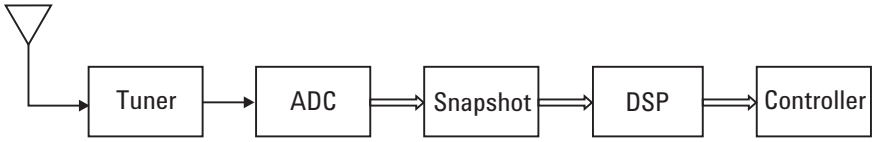
Example

The following command defines a snapshot module in the `e3238s.cfg` file:

```
searchRx1.snapshotMemory1.model:          E9830A
searchRx1.snapshotMemory1.interfaceParm: 133
```

The example above defines the configuration of one E9830A module as shown in the block diagram in figure 52. Because it's input is the VXI Local Bus, it must be installed between the ADC and the DSP modules in the VXI chassis.

Figure 52. Snapshot memory block diagram



See Also

searchRx.snapshotMemory.interfaceParm (page 150)

searchRx.switchConfiguration

Syntax `searchRx1.switchConfiguration: string`

`The maximum length is 31 characters.`

Description This command is used to program the VXI multiplex switch. It specifies the string to appear in the Switch Configuration dialog with the label "Configuration".

Example The following command represents a switch configuration with 16 inputs (from antennas) and 1 output.

```
searchRx1.switchConfiguration: 16 : 1
```

searchRx.switchConnections

Syntax

```
searchRx1.switchConnections: string

The maximum length is 255 characters.
```

Description

This command is used to program the E1472A RF multiplex switch.

Specifies labels that appear in the Show Connections dialog box. These are used to identify what each connector should be connected to.

Example

The following example contains variables (@A1, @A2, ...) that are replaced automatically when an antenna name is defined.

```
searchRx1.switchInterfaceParm: 131
searchRx1.switchConfiguration: 16 : 1
searchRx1.switchConnections:   @A1,@A2,To 40,@A3,@A4,\
                               @A5,@A6,To 41,@A7,@A8,\
                               @A9,@A10,To 42,@A11,@A12,\
                               @A13,@A14,To 43,@A15,@A16,\
                               From Com 00,From Com 10,To Tuner,\
                               From Com 20,From Com 30,\
                               ,,,,,,,,
```

Note

This example show the lines indented so that it's easy to read. If spaces are used, you can exceed the 255 character limit. Anything after the 255th character will not appear in the displayed wiring dialog box.

searchRx.switchInterfaceParm

Syntax `searchRx1.switchInterfaceParm: integer`

`1 ≤ integer ≤ 254`

Description Specifies the VXI logical address of the RF switch module.

The range specified above applies to VXI logical addresses. The address must correspond with switches set inside the module.

Example `searchRx1.switchInterfaceParm: 131`

See Also The Hardware Installation section of this book.

searchRx.switchModel

Syntax

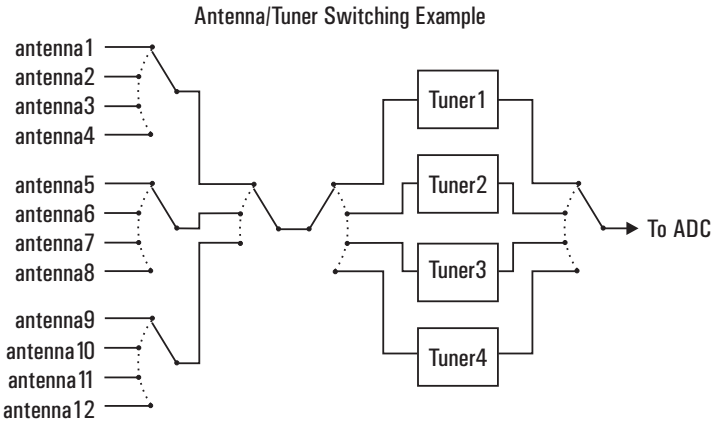
```
searchRx1.switchModel: enum

enum = {None, E1472A, E1368/69/70A}
```

Description

Specifies the RF switch model installed.

A switch may be used to implement multiple antennas and tuners. Figure 53 is an example configuration showing multiple antennas and multiple tuners. See searchRx.tuner.tunerModel (page 157).

Figure 53. Tuner switching diagram



Only one switch module may be used at a time. The following models are supported.

- **E1472A** is a 50Ω RF switch with 6 4:1 blocks to which additional blocks may be added.
- **E1474A** is a 75Ω RF switch with 6 4:1 blocks to which additional blocks may be added.
- **E1368A** is a microwave switch that provides as many as 5 switches (3 internal and 2 external) for signal frequencies from DC to 18 GHz.
- **E1369A** is a microwave switch for which individual switch relays may be selected and installed to provide as many as 5 switches (3 internal and 2 external) for signal frequencies from DC to 26.5 GHz.
- **E1370A** is a microwave switch/step attenuator that provides either a single-pole, multithrow switch or a programmable step attenuator.

searchRx.tuner.tunerInterfaceParm

Syntax

```
searchRx1.tuner(1..4).tunerInterfaceParm: string

The maximum length is 39 characters.
```

Description

Specifies the tuner interface parameter(s). This varies from one type of tuner to another. It consists of the information required to communicate with the tuner and, in some cases, an operating configuration such as which LO outputs are active. See searchRx.tuner.tunerModel (page 157) for the list of tuner models.

Tuner Parameters

Each tuner type has unique parameters defined in the d.e3238s.cfg file as follows:

- **WJ9119-1**: An HF tuner comprised of two VXI modules (see pg 47)

```
searchRx1.tuner1.tunerModel:           WJ9119-1
searchRx1.tuner1.tunerInterfaceParm:   140, 142, 0, 0, 1, 1,
500, 500
```

  The parameters are defined in order as given:

  - LO module logical address 1 to 254
  - RF module logical address 1 to 254
  - 10 MHz Reference 0 (Internal) or 1 (External)
  - ADC Clock Output 1 to 255
  - 1st LO Output 1 to 255
  - 2nd LO Output 1 to 255
  - settle time (steps < 7.5 MHz) in microseconds, default is 500
  - settle time (steps > 7.5 MHz) in microseconds, default is 500

  "Output" parameters specify which outputs are enabled on the LO front panel. The valued entered is a mask for which outputs are enabled. For example:

  - 1: Enable output 1.
  - 7: Enable output 1, 2, and 3.
  - 255: Enable all outputs.

- **E2730B/E2731B**: A VHF/UHF tuner that is a VXI module (see page 48)

```
searchRx1.tuner1.tunerModel:           E2731B
searchRx1.tuner1.tunerInterfaceParm:   136, 0, 3, 20
```

  The parameters are defined in order as given:

  - Logical Address 1 to 255
  - 10 MHz Reference 0 (Internal), 1 (External) or 2 (VXI Backplane)
  - Settling Time 1 to 1000 mSec Typical Range: 3 to 5 mSec
  - Start Frequency 2 MHz to 20 MHz Default: 20 MHz
  - 89605 Logical Addr Optional, 1 to 255

  The adcModel can be E1439x/70 or N6830A/70.

- **N6830/HF**: A dual channel HF receiver and ADC

```
searchRx1.tuner1.tunerModel:        N6830A/HF
searchRx1.tuner1.tunerInterfaceParm:  1 to 255
```

  Channel 1 uses the logical address selected by the switch setting, channel 2 is this address plus one. When the tuner type is N6830A/HF, the 70 MHz IF input is not used. The search receiver ADC model must be specified as N6830A/HF. See searchRx.adcModel (page 138)

searchRx.tuner.tunerModel

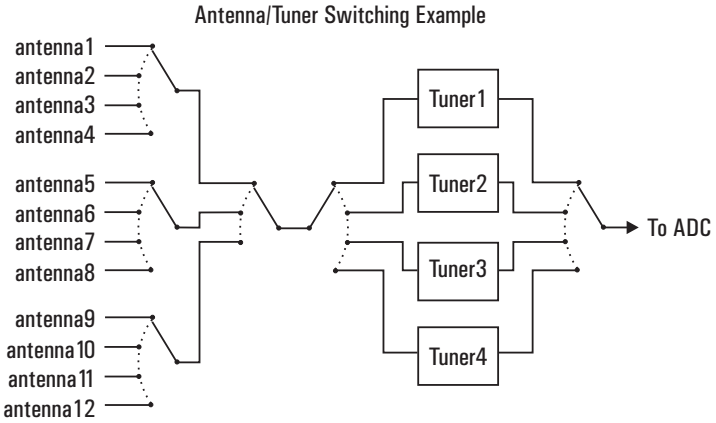
Syntax

```
searchRx1.tuner(1..4).tunerModel: enum

enum = {None, HP89430A, HP89431A, WJ9119, WJ9119-1, CS5040, Interad9640,
CS5320A, E2730A, E2731A, E2730B, E2731B, SI9250-ADV3000, SI9250-E273X,
SI9250, ADV3000, PSA, SI9136, SI9250-SI9136, N6830/HF, Off }
```

Description

Specifies the tuner model(s) for the *search* receiver. Multiple tuners may be specified. A switch is used to implement multiple antennas and tuners. Figure 54 is an example configuration showing multiple antennas and multiple tuners. See searchRx.antenna.switchCmnd (page 141).

Figure 54. Tuner switching diagram



See page 11 for a general discussion of tuners.

Examples

To specify one tuner, use the following syntax:

```
searchRx1.tuner1.tunerModel:          WJ9119-1
searchRx1.tuner1.tunerInterfaceParm:  140, 142, 0, 0, 1, 1
```

To specify multiple tuners, use the following syntax:

```
searchRx1.tuner1.tunerModel:          None
searchRx1.tuner2.tunerModel:          WJ9119-1
searchRx1.tuner2.tunerInterfaceParm:  140, 142, 0, 0, 1, 1
searchRx1.tuner3.tunerModel:          HP89431A
searchRx1.tuner3.tunerInterfaceParm:  125000
searchRx1.tuner4.tunerModel:   Off
```

Models Supported

- **None** is baseband operation which uses the span of the ADC.
- **89431A** is a non-VXI, 2-2,650 MHz tuner with an RS-232 interface.
- **WJ9119-1** is a VXI, 0.5-32 MHz tuner, IF BW = 8 MHz, use with E1437A ADC.
- **E2730A/B** is a VXI, 20-2700 MHz tuner; use with E1439A/B/D ADC, only.
- **E2731A/B** is a VXI, 20-6000 MHz tuner; use with E1439A/B/D ADC, only.
- **CS5040A** is a VXI, 20-18,000 MHz tuner; use with E1439A/B/D ADC, only.
- **SI9250** is a VXI block downconverter, use with E2730B tuner and E1439D ADC.
- **SI9136B** is a Dual Channel VXI Digital VME Tuner, requires VXI carrier module.
- **PSA** is a signal analyzer instrument. See page 23.
- **N6830A/HF** Dual Channel HF receiver, 0.2 - 32 MHz, only with N6830A/HF ADC.
- **Off** is used in a multiple-tuner configuration with less than 4 tuners installed.

Note

In directed search, one tuner must be used in all bands. Different antennas can be used for each band but the tuner selection cannot be changed during a sweep.

searchRx.tuner.tunerSwitchCmnd

Syntax `searchRx1.tuner(1..4).tunerSwitchCmnd: integer`

Description Specifies the switch relay settings necessary to select a certain tuner.

The value is logically compared with the searchRx.tuner.tunerSwitchMask (page 159) command to determine which relays need to change. Minimizing relay activity maximizes switch contact life.

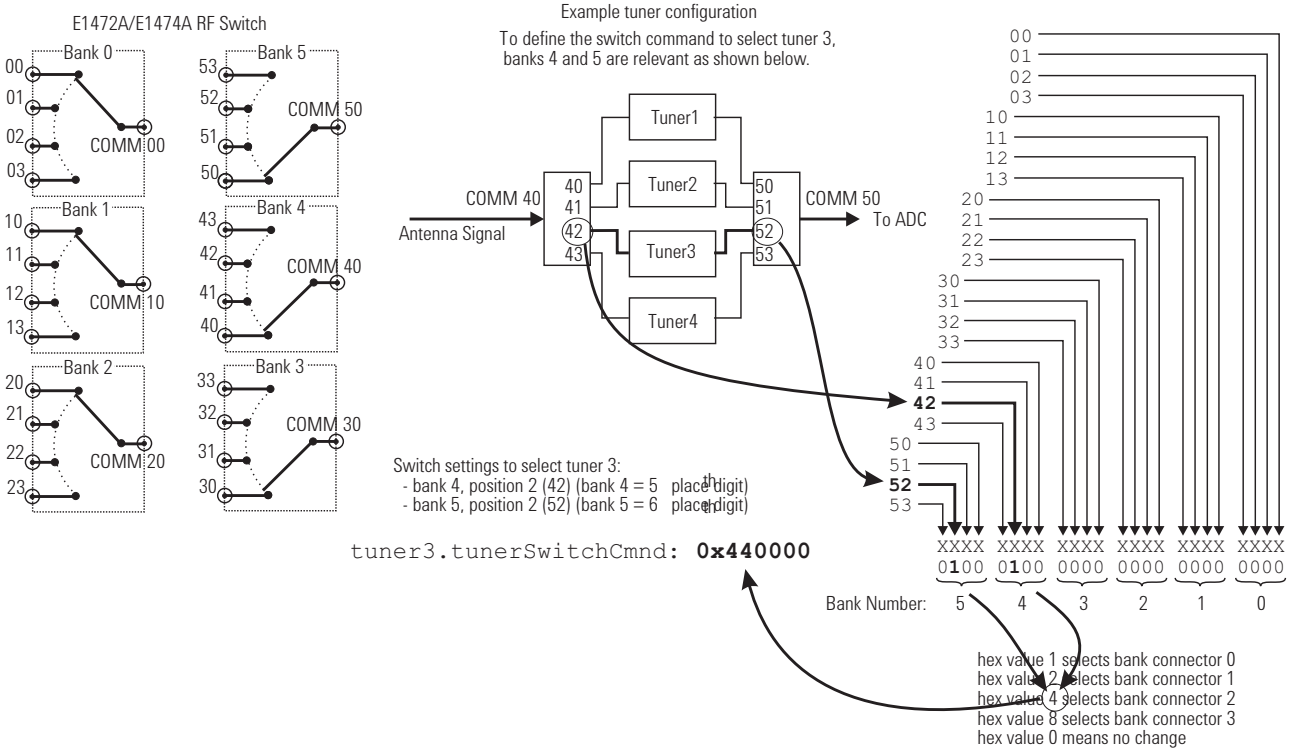
The value entered may be either an integer or hexadecimal value.

This command supports use of multiple tuners.

Example

```
searchRx1.tuner1.tunerSwitchCmnd: 0x110000
searchRx1.tuner1.tunerSwitchMask: 0xFF0000
searchRx1.tuner2.tunerSwitchCmnd: 0x220000
searchRx1.tuner2.tunerSwitchMask: 0xFF0000
```

Figure 55. Tuner switch command setting



searchRx.tuner.tunerSwitchMask

Syntax `searchRx1.tuner(1..4).tunerSwitchMask: integer`

Description This parameter value is used in conjunction with searchRx.tuner.tunerSwitchCmnd (page 158) to limit the number of switch relay changes needed to select a specified tuner. This maximizes the switch contact reliability and life.

Example

```
searchRx1.tuner1.tunerSwitchCmnd: 0x110000
searchRx1.tuner1.tunerSwitchMask: 0xFF0000
searchRx1.tuner2.tunerSwitchCmnd: 0x220000
searchRx1.tuner2.tunerSwitchMask: 0xFF0000
```

searchRx.tuner.userCalFile

Syntax

```
searchRx1.tuner(1..16).userCalFile: filename
The maximum length is 79 characters.
```

Description Specifies the filename of the calibration data to be used for a given tuner.

Example

```
!These commands control the application of corrections to compensate for
!tuner response and antenna path response. These user-supplied
!corrections are in addition to built-in RF and IF corrections. For an
!example of the file format, see the file e3238s\cal\d.tuner1.cal.
!Cal files are normally located in the C:\E3238s\cal
!searchRx1.tuner1.userCalFile:  tuner1.cal
```

searchRx.AdcSynchronization (EMC)

Syntax `searchRx.AdcSynchronization: enum`

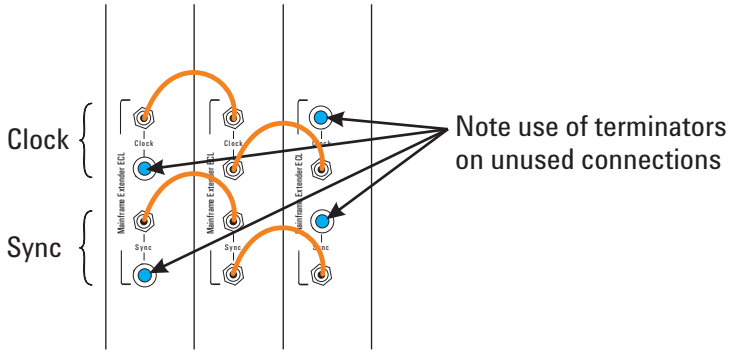
`enum = {VXIBackplane, ADCFrontPanel}`

Description This specifies the routing of the master ADC's clock and sync signals are passed from the master ADC to the slave ADCs.

`VXIBackplane` specifies that the signals are passed on bus lines at the rear of the mainframe.

`ADCFrontPanel` specifies that the signals are passed through the ADC front-panel connectors as shown in figure 56.

Figure 56. ADC front-panel clock and sync connections



searchRxConfiguration (EMC)

Syntax `searchRxConfiguration: enum`

`enum = {SingleChannelSearch, MultiChannelParallelTimeSynchronousSearch, MultiChannelParallelPhaseSynchronousSearch}`

Description Specifies how the search receiver section operates:

`SingleChannelSearch` is the common operation type in which one channel sweeps.

`MultiChannelParallelTimeSynchronousSearch` is the mode in which multiple search receiver channels sweep the same regions and sampling is time-synchronous.

`MultiChannelParallelPhaseSynchronousSearch` is the mode in which multiple search receivers sweep the same regions and the measurement results are phase synchronous. Note: this mode requires calibration functionality that is not implemented.

signal.alias

Syntax

```
signal(1..32).alias: string
```

The maximum length is 31 characters.

Description Specifies an alternative name to be used in the application's user interface so that the real signal name is not displayed.

Example

```
multipleSignalsPerProcessor: Enabled

asxDsp_0:   C:\E3238s\downloadables\ASXdsp_0.elf
asxDsp_1:   C:\E3238s\downloadables\ASXdsp_1.elf

!
! Demo Signal Type
!

signal1.enabled:     False
signal1.hostLib:     C:\E3238S\demo\demoHost.dll
signal1.hostDsp:     C:\E3238S\demo\demoDsp.dll
signal1.targetDsp:   C:\E3238S\demo\demoDsp.esl
signal1.loadFactor:  64
signal1.minChannels: 4
signal1.maxChannels: 32
signal1.args:        21 45 A friday
signal1.alias:       Bob
```

Note You may use either regular slashes (/) or back slashes (\) in the pathname.

signal.args

Syntax

```
signal(1..32).args: string
The maximum length is 255 characters.
```

Description Specifies the values for any parameters that a custom library might pass in.

Example

```
multipleSignalsPerProcessor: Enabled
asxDsp_0:   C:\E3238s\downloadables\ASXdsp_0.elf
asxDsp_1:   C:\E3238s\downloadables\ASXdsp_1.elf
!
! Demo Signal Type
!
signal1.enabled:     False
signal1.hostLib:     C:\E3238S\demo\demoHost.dll
signal1.hostDsp:     C:\E3238S\demo\demoDsp.dll
signal1.targetDsp:   C:\E3238S\demo\demoDsp.esl
signal1.loadFactor:  64
signal1.minChannels: 4
signal1.maxChannels: 32
signal1.args:        21 45 A friday
signal1.alias:       Bob
```

Note You may use either regular slashes (/) or back slashes (\) in the pathname.

signal.enabled

Syntax

```
signal(1..32).enabled: enum
enum = {False, True}
```

Description

Specifies whether the signal processing components are to be loaded. This allows you to disable a signal library without commenting out all the lines associated with it.

Another way to disable a signal library is to leave the setting True and just comment out the signal.enabled line.

Example

The following example shows a signal that is disabled:

```
multipleSignalsPerProcessor: Enabled

asxDsp_0:   C:\E3238s\downloadables\ASXdsp_0.elf
asxDsp_1:   C:\E3238s\downloadables\ASXdsp_1.elf

!
! Demo Signal Type
!

signal1.enabled:     False
signal1.hostLib:     C:\E3238S\demo\demoHost.dll
signal1.hostDsp:     C:\E3238S\demo\demoDsp.dll
signal1.targetDsp:   C:\E3238S\demo\demoDsp.esl
signal1.loadFactor:  64
signal1.minChannels: 4
signal1.maxChannels: 32
signal1.args:
signal1.alias:
```

Note

You may use either regular slashes (/) or back slashes (\) in the pathname.

signal.hostDsp

Syntax

```
signal(1..32).hostDsp: filename
The maximum length is 127 characters.
```

Description Specifies the filename of the DSP library to be loaded in the host (system controller). This provides the same signal processing normally performed in the G4 processor for the case where you run the E3238S application in the -noHardware mode.

Example

```
multipleSignalsPerProcessor: Enabled

asxDsp_0:   C:\E3238s\downloadables\ASXdsp_0.elf
asxDsp_1:   C:\E3238s\downloadables\ASXdsp_1.elf

!
! Demo Signal Type
!

signal1.enabled:     False
signal1.hostLib:     C:\E3238S\demo\demoHost.dll
signal1.hostDsp:     C:\E3238S\demo\demoDsp.dll
signal1.targetDsp:   C:\E3238S\demo\demoDsp.esl
signal1.loadFactor:  64
signal1.minChannels: 4
signal1.maxChannels: 32
signal1.args:        21 45 A friday
signal1.alias:       Bob
```

Note You may use either regular slashes (/) or back slashes (\) in the pathname.

signal.hostLib

Syntax

```
signal(1..32).hostLib: filename
The maximum length is 127 characters.
```

Description Specifies the filename of the library to be loaded in the host (system controller). This provides the host component of the typical host-target interaction mechanism.

Example

```
multipleSignalsPerProcessor: Enabled

asxDsp_0:   C:\E3238s\downloadables\ASXdsp_0.elf
asxDsp_1:   C:\E3238s\downloadables\ASXdsp_1.elf

!
! Demo Signal Type
!

signal1.enabled:     False
signal1.hostLib:     C:\E3238S\demo\demoHost.dll
signal1.hostDsp:     C:\E3238S\demo\demoDsp.dll
signal1.targetDsp:   C:\E3238S\demo\demoDsp.esl
signal1.loadFactor:  64
signal1.minChannels: 4
signal1.maxChannels: 32
signal1.args:        21 45 A friday
signal1.alias:       Bob
```

Note You may use either regular slashes (/) or back slashes (\) in the pathname.

signal.loadFactor

Syntax `signal(1..32).loadFactor: integer`

`1 ≤ integer ≤ 96`

Description Specifies the maximum number of narrow-band channels that one G4 processor can support for a particular signal type. This value is determined during the development of the processing library. It is used to determine the processing topology, given how many other signals are loaded, their load factors, and the signal processing hardware assets installed.

Example The following example shows a load factor of 64 channels per processor.

```
multipleSignalsPerProcessor: Enabled

asxDsp_0:   C:\E3238s\downloadables\ASXdsp_0.elf
asxDsp_1:   C:\E3238s\downloadables\ASXdsp_1.elf

!
! Demo Signal Type
!

signal1.enabled:     False
signal1.hostLib:     C:\E3238S\demo\demoHost.dll
signal1.hostDsp:     C:\E3238S\demo\demoDsp.dll
signal1.targetDsp:   C:\E3238S\demo\demoDsp.esl
signal1.loadFactor:  64
signal1.minChannels: 4
signal1.maxChannels: 32
signal1.args:        21 45 A friday
signal1.alias:       Bob
```

Note You may use either regular slashes (/) or back slashes (\) in the pathname.

signal.maxChannels

Syntax

```
signal(1..32).maxChannels: string
The maximum length is 39 characters.
```

Description

Specifies the maximum number of narrow-band signal processing channels supported by the given signal type.

The theoretical maximum is 928 channels (9 * 96) + 64.[1]

When signal libraries are loaded, the application determines how the channels and processors are allotted based on the number of signal processing libraries loaded and the various channel requirements associated with each, as well as the number of DDC channels and G4 processors that are available in the installed hardware.

Example

The following example shows a maximum number of channels of 32.

```
multipleSignalsPerProcessor: Enabled

asxDsp_0:   C:\E3238s\downloadables\ASXdsp_0.elf
asxDsp_1:   C:\E3238s\downloadables\ASXdsp_1.elf

!
! Demo Signal Type
!

signal1.enabled:     False
signal1.hostLib:     C:\E3238S\demo\demoHost.dll
signal1.hostDsp:     C:\E3238S\demo\demoDsp.dll
signal1.targetDsp:   C:\E3238S\demo\demoDsp.esl
signal1.loadFactor:  64
signal1.minChannels: 4
signal1.maxChannels: 32
signal1.args:        21 45 A friday
signal1.alias:       Bob
```

Note

You may use either regular slashes (/) or back slashes (\) in the pathname.

[1]The E3238S supports as many as ten E9821A modules. Each module has four sites so the configuration supporting the maximum channel count is three 32-channel DDCs and one G4 processor. At least one dual-G4 card is required for search processing on the first module so it can provide 64 channels.

signal.minChannels

Syntax

```
signal(1..32).minChannels: string
The maximum length is 39 characters.
```

Description Specifies the minimum number of narrow-band signal processing channels supported by the given signal type.

When signal libraries are loaded, the application determines how the channels and processors are allotted based on the number of signal processing libraries loaded and the various channel requirements associated with each, as well as the number of DDC channels and G4 processors that are available in the installed hardware.

Example The following example shows a minimum number of channels of 4.

```
multipleSignalsPerProcessor: Enabled

asxDsp_0:   C:\E3238s\downloadables\ASXdsp_0.elf
asxDsp_1:   C:\E3238s\downloadables\ASXdsp_1.elf

!
! Demo Signal Type
!

signal1.enabled:     False
signal1.hostLib:     C:\E3238S\demo\demoHost.dll
signal1.hostDsp:     C:\E3238S\demo\demoDsp.dll
signal1.targetDsp:   C:\E3238S\demo\demoDsp.esl
signal1.loadFactor:  64
signal1.minChannels: 4
signal1.maxChannels: 32
signal1.args:        21 45 A friday
signal1.alias:       Bob
```

Note You may use either regular slashes (/) or back slashes (\) in the pathname.

signal.targetDsp

Syntax

```
signal(1..32).targetDsp: filename
The maximum length is 127 characters.
```

Description

Specifies the file to be downloaded to the target (G4) processor card.

This file supports the ELF shared library (ESL) usage in which a single G4 processor can handle multiple signal types.

When this file is used, the `targetDsp_0` and `targetDsp_1` commands are not used.

Example

```
multipleSignalsPerProcessor: Enabled

asxDsp_0:   C:\E3238s\downloadables\ASXdsp_0.elf
asxDsp_1:   C:\E3238s\downloadables\ASXdsp_1.elf

!
! Demo Signal Type
!

signal1.enabled:     False
signal1.hostLib:     C:\E3238S\demo\demoHost.dll
signal1.hostDsp:     C:\E3238S\demo\demoDsp.dll
signal1.targetDsp:   C:\E3238S\demo\demoDsp.esl
signal1.loadFactor:  64
signal1.minChannels: 4
signal1.maxChannels: 32
signal1.args:        21 45 A friday
signal1.alias:       Bob
```

See Also

multipleSignalsPerProcessor (page 132)

asxDsp_0 (page 92)

Note

You may use either regular slashes (/) or back slashes (\) in the pathname.

socketServer

Syntax `socketServer: enum`

`enum = {Disabled, Enabled}`

Description Specifies whether the sockets feature is active.

The default value of this parameter is *Disabled*.

Example The following commands show an example socket configuration:

```
socketServer:                Enabled
maxServices:                 1
maxClientSockets:            0
socketServerTimerInterval:   5
```

See Also
maxClientSockets (page 124)
maxServices (page 125)
socketServerTimerInterval (page 173)
e3238sService (page 104)
e3238sServiceDataBufferSize (page 105)
e3238sServiceMaxConnections (page 106)
e3238sServiceRecvBufferSize (page 107)
e3238sServiceSendBufferSize (page 108)

socketServerTimerInterval

Syntax `socketServerTimerInterval: integer`

`1 ≤ integer ≤ 10000`

Description This specifies how often the system checks for the presence of sockets data.

Units are milliseconds.

Default value is 5 ms.

Example The following commands show an example socket configuration:

```
socketServer:                Enabled
maxServices:                 1
maxClientSockets:            0
socketServerTimerInterval:   5
```

See Also
maxClientSockets (page 124)
maxServices (page 125)
socketServer (page 172)
e3238sService (page 104)
e3238sServiceDataBufferSize (page 105)
e3238sServiceMaxConnections (page 106)
e3238sServiceRecvBufferSize (page 107)
e3238sServiceSendBufferSize (page 108)

timeCorrectionMaxBlocksize

Syntax `timeCorrectionMaxBlocksize: integer`

`1 ≤ integer ≤ 2147483647`

Description Specifies the maximum block size for applying Time Corrections.

When Time Correction processing is enabled, time snapshot files larger than this value will not be corrected. If not specified, this defaults to 16 Mega-samples.

Time correction processing is performed on the host computer and is memory intensive, requiring 8 bytes for each sample.

This value can be increased if large corrected capture files are needed, but the host computer should have adequate physical memory available.

timeReference

Syntax `timeReference: enum`

`enum = {SystemClock, VXIIRIG}`

Description Specifies whether to use the system clock or an IRIG time reference module as the time reference source. This is the reference for all time stamp values.

Example

```
timeReference:                vxiIRIG
irigModel:                    bc350vxi
irigInterfaceParm:            200
irigOperatingMode:            Decode
irigTimeCodeFormat:           IRIGB
irigTimeCodeModulation:       AM
irigClock:                    Internal
irigSecondsFromGMT:           0
irigCableDelay:               0
```

See Also irigModel (page 119)

timeReferenceCal

Syntax `timeReference: integer`

`integer = [0, 1]`

Description Enables adjustment of ADC sample clock generated timestamps to track IRIG time. Only applies with tuner locked and vxiIRIG timeReference

Example

```
timeReference:                  vxiIRIG
timeReferenceCal                0
irigModel:                      bc350vxi
irigInterfaceParm:              200
irigOperatingMode:              Decode
irigTimeCodeFormat:             IRIGB
irigTimeCodeModulation:         AM
irigClock:                      Internal
irigSecondsFromGMT:             0
irigCableDelay:             0
```

timerInterruptInterval

Syntax `timerInterruptInterval: integer`

`1 ≤ integer ≤ 999`

Description This time value specifies how often tasks associated with external resources are performed. These include managing handoff receivers and narrow-band signal processing channels.

When a signal processing library is loaded, this specifies how often (in milliseconds) the E3238S system is interrupted to handle messaging between the host and target processes.

The default is 100 ms.

Great care should be used in selecting this value. Values too small inhibit the search process (E3238S) and values too large may cause the signal processing message buffer to overflow.

userAlarmTask

Syntax `userAlarmTask: filename`

`The maximum length is 511 characters.`

Description Specifies the user-defined alarm task (shared-library) program to load.

Example `userAlarmTask: C:\e3238s\userTaskEmail.dll`

Note You may use either regular slashes (/) or back slashes (\) in the pathname.

userMenu

Syntax `userMenu: filename`

`The maximum length is 511 characters.`

Description Specifies the user-defined menu shared-library program to load.

As many as 4 user-defined menu bar entries may be defined, each with as many as 8 menu entries.

Example

```
userMenu: C:/e3238s/examples/userMenu/userMenu.dll,\
          C:/e3238s/examples/userMenu/userMenuArrow.dll
```

Note You may use either regular slashes (/) or back slashes (\) in the pathname.

userPane

Syntax `userPane: filename`

`The maximum length is 511 characters.`

Description Specifies the user-defined shared library that implements a custom window pane.

The maximum number of user-defined panes allowed is 4.

To display a pane, add it to the application window with the `layout.pane.type` command.

Example `userPane: C:/e3238s/examples/userPane/userPane.sl`

Note You may use either regular slashes (/) or back slashes (\) in the pathname.

userThreshold

Syntax `userThreshold: filename`

`The maximum length is 511 characters.`

Description Specifies the user-defined shared library that implements a custom threshold algorithm.

vxiInterface

Syntax `vxiInterface: string`

`The maximum length is 31 characters.`

Description This command identifies the VISA[1] name for the VXI system as it is defined in the controller software.

See Configuring the VXI interface (page 70)

The default name is `VXI0` (zero).

[1]VISA (Virtual Instrument Software Architecture) is an I/O library that provides a programming interface to VXI*plug&play* instrument drivers.

Application Resources

X Window applications can declare variables such that their values may be set in an external ASCII file. These settings are called *resources*. The `E3238s` file is an application *resource* file. The entries in this file define fonts, colors, line thickness, window sizes, file lengths, path names, and many other things for each part of the `e3238s` program. Other application parameters defined in this file are:

- Application Resource file used (see Appendix A: d.E3238s Listing (pg 245))
- Hardware configuration file listing (see Appendix B: d.e3238s.cfg Listing (pg 247)
- Function key definitions
- Accelerator key definitions
- Memory usage parameters
- Capacity of the log views
- Titles of trace panes and the handoff log pane

This allows you to change how the program looks and works without changing a program source file and recompiling. For more information about Motif resources, see one of the many Motif references available such as *X Toolkit Intrinsics Programming Manual*, Volume Four of the X Window Series by O'Reilly & Associates.

When you first install the software, no resource file exists. Instead, a file named `d.E3238s` is created in the `e3238s` directory ("d" is for default). This avoids overwriting an existing file when the "installation" is really an update. So, after performing an installation on a new disk, you may need to copy this file to create a file named `E3238s`.

Custom Resources

When the program starts it uses the first resource file it finds; the order of the search is as follows:

1. User Profile directory
2. User HOME directory
3. E3238s directory (as defined during installation)
4. Standard Exceed locations (e.g., XUSERFILESEARCHPATH)

If no E3238s file is found, fallback resources defined in the program are used. See also, the discussion above.

Any values specified on the command line with the `-xrm` option are loaded for that instance of the program and override any conflicting variable settings specified in the resource files.

You can specify a resource when you start the program by using the `-xrm` option as follows:

```
e3238s -xrm resourcestring
```

This option specifies a resource name and value to override any defaults. It is also very useful for setting resources that don't have explicit command line arguments.

It is recommended that you start by copying the E3238s file into your home directory. This allows users to have custom configurations.

Widget resources, such as fonts and colors, can be set for most widgets in this software. The widget hierarchy can be printed from the File menu.

Application resources are those resources created specifically for this application and are not part of the OSF/Motif widget set. If these resources are not specified, their default values are used. The application resources are described below:

alarmLogViewLength type: Integer default: 100

Specifies the number of alarm entries to keep in the alarm log view. Each handoff requires approximately 90 bytes of memory.

Example: `*alarmLogViewLength: 400`

audioOutput type: XmRString default: 0

Specifies the location of the audio output. The default is the same location that the e3238s code is executing on.

clientTitles type: Boolean default: False

Specifies whether to put titles on dialog boxes.

commandHelpVolume type: XmRString

default:

```
*commandHelpVolume: <E3238s_home>/help/CommandRef
```

Specifies the help volume for E3238s commands.

Note Windows® path names must use either / or \\ character delimiters.

commandLineEnabled type: Boolean default: True

Specifies whether to enable the command line pane.

Example: `*commandLineEnabled: True`

commandLineHistoryLength type: Integer default: 100

Specifies the number of commands to keep in the command line history. Each command requires approximately 80 bytes of memory.

Example: `*commandLineHistoryLength: 300`

dataBufferSize type: Integer default: 8,000,000

Specifies the amount of memory to allocate for data storage in terms of data points. Each data point is 4 bytes. To sweep from 2 MHz to 2.65 GHz with a 940 Hz RBW, the amount of memory required is calculated as follows:

1. Find the bin spacing. The following tables show the relationship between shape factor, bin spacing, and RBW[1].

RBW (Hz) vs. Bin Spacing and Shape Factor

E1437 and N6830A/HF ADC
20.48 MSamples/Sec and 81.92 MSamples/Sec

	Shape Factor		
Bin Spacing	9.0:1	4.0:1	2.6:1
80,000.0	120 k	177 k	305 k
40,000.0	60 k	88 k	152 k
20,000.0	30 k	44 k	76 k
10,000.0	15 k	22 k	38 k
5,000.0	7.5 k	11 k	19 k
2,500.0	3.8 k	5.5 k	9.6 k
1,250.0	1.9 k	2.8 k	4.8 k
625.0	940	1.4 k	2.4 k
312.5	470	700	1.2 k
156.25	240	350	600
78.125	120	180	300
39.0625	60	90	150
19.5313	29	43	80
9.7656	15	22	37
4.8828	7.3	11	19

E1438 ADC
102.4 MSamples/Sec

	Shape Factor		
Bin Spacing	9.0:1	4.0:1	2.6:1
800,000.0	1.2 M	1.8 M	3.1 M
400,000.0	600 k	886 k	1.5 M
200,000.0	300 k	443 k	764 k
100,000.0	150 k	222 k	382 k
50,000.0	75 k	111 k	191 k
25,000.0	37.5 k	55 k	95 k
12,500.0	18.8 k	28 k	48 k
6,250.0	9.4 k	14 k	24 k
3,125.0	4.7 k	6.9 k	12 k
1,562.5	2.4 k	3.5 k	6 k
781.25	1.2 k	1.7 k	3 k
390.625	590	870	1.5 k
195.3125	300	440	750
97.6563	150	220	380
48.8281	80	110	190
24.4141	37	60	100
12.2070	18	27	50
6.1035	9.2	14	23

E1439/BB, E1439/70, N6830/70 ADC
95 MSamples/Sec

	Shape Factor		
Bin Spacing	9.0:1	4.0:1	2.6:1
742,187.5	1.1 M	1.6 M	2.8 M
371,039.75	556 k	822 k	1.4 M
185,546.875	278 k	411 k	708 k
92,773.4375	139 k	205 k	354 k
46,386.7187	69.6 k	102 k	177 k
23,193.3594	34.8 k	51.4 k	88.6 k
11,586.6797	17.4 k	25.7 k	44.3 k
5,798.3398	8.7 k	12.8 k	22.1 k
2,899.1699	4.3 k	6.4 k	11.1 k
1,449.5850	2.2 k	3.2 k	5.5 k
724.7925	1.1 k	1.6 k	2.8 k
362.3962	550	810	1.4 k
181.1981	280	410	700
90.5991	140	210	350
45.2995	70	110	180
22.6498	34	60	90
11.3249	17	25	43
5.6624	8.5	13	22

[1]When running narrowband processing, such as narrowband recorder, the low end of the RBW setting is limited. For example, an E1439 with a 9.0:1 shape factor would have a low end RBW of 140 Hz.

2. Next, calculate the number of frequency points.

   ```
   num points = (Stop Frequency - Start Frequency)/ Bin Spacing
   ```

   For our example this would be:

   ```
   num points = (2.65 GHz - 2 MHz) / 625
   num points = 4,236,800
   ```

3. Now, calculate the host computer memory required.

   ```
   num data points = num points × bytes required per data point
   ```

   For our example this would be:

   ```
   bytes = 4,236,800 × 4
   bytes = 16,947,200
   ```

For this example, almost 17 MB of RAM is required. The value you would assign to the resource is in points.

Example: `*dataBufferSize: 4,300,000`

The software fails when it cannot allocate the amount of memory you specify.

defaultAudioExtension type: XmRString default: *.wav

Specifies the default extension for audio files. The possible audio extensions are:

```
*.u    MuLaw (u-law)
*.al   ALaw (A-law)
*.au   Sun (NeXT)
*.wav  Microsoft RIFF waveform
*.snd  Next
*.l16  Linear16 (16-bit signed)
*.l8   Linear8 (8-bit signed)
*.lo8  Linear8Offset (8-bit unsigned)
```

displayLocalTime type: Boolean default: True

Specifies whether time information is displayed using the local timezone information. When False, time values are displayed as GMT.

e3238sIconPixmap type: XmRString default: none

Specifies a pixmap file to be used as an icon.

enableAudio type: Boolean default: True

Specifies whether the audio output is enabled.

enhancedSpectrogramMarker type: Boolean default: True

Enables the enhanced spectrogram marker.

enhancedSpectrumMarker type: Boolean default: False

Enables the enhanced spectrum marker that adds time information to the marker information.

gridBitmap type: XmRString

default:

```
*gridBitmap:  <E3238S_home>/bitmaps/grid.bm
```

Specifies the grid bitmap used when displaying the handoff frequencies. The handoff frequency is displayed as a solid line and, as bandwidth increases, the trace area is filled with this bitmap. See following note.

hardwareConfiguration type: XmRString

default:

```
<E3238S_home>/e3238s.cfg
```

Specifies the hardware configuration file. See note below.

Example: `*hardwareConfiguration:  /E3238s/e3238s.cfg.mine`

You can also start the program with the -xrm flag and specify this file. This is very useful for specifying multiple startup icons, each with a different configuration file and/or initial state (specified with the -missionState flag).

handoffLogViewLength type: Integer default: 500

Specifies the number of handoffs to keep in the handoff log view. Each handoff requires approximately 160 bytes of memory.

handoffPaneFont type: XmRString default: 7x14

Specifies the font used by spreadsheet area of the handoff receiver pane.

hideDisplay type: Boolean default: False

Prevents the software from displaying an X window. Error messages *are* displayed. See also remoteMode (pg 189).

help4helpVolume type: XmRString default: Help4Help

Specifies the help volume that provides help for help.

helpVolume type: XmRString

default:

```
<E3238S_home>/help/e3238s
```

Specifies the help volume.

Note

Windows® path names must use either / or \\ character delimiters.

logViewFont type: XmRString default: 7x14

Specifies the fonts for the log views. A fixed spaced font should be used.

mainLoopSelectTime type: Integer default: 400

The main loop of this software continually checks the VXI hardware for changes in status. This method uses a large amount of the controller's processing power. This resource, in microseconds, specifies the amount of time this process is to suspend, and allow other processes to run, each cycle through the main loop when a sweep is not occurring. This is implemented through the *select* function call. While a sweep is active, the software does not call the *select* function in the main loop.

maxEnergyHistorySize type: Integer default: 5000

Specifies the maximum number of entries the energy history can contain. When this limit is reached, no new entries can be added to the energy history until some are deleted or the entire energy history is cleared. Each entry is about 128 bytes.

maxHandoffRxs type: Integer default: 16

Specifies the maximum number of handoff receivers that can be controlled at one time. The maximum number is 100.

maxSpectrogramColors type: Integer default: 32

Specifies the maximum number of colors cells to allocate for the color spectrogram display. The maximum allowable is 128.

multiClickTime type: integer default: 200

Specifies the mouse double-click time in milliseconds.

newEnergyLogViewLength type: Integer default: 1000

Specifies the number of new energy entries to keep in the new energy log view. Each handoff requires approximately 60 bytes of memory.

openCommandPort type: String default: none

Specifies a command port program to run at power up.

openScreenTime type: Integer default: 5

Specifies the amount of time the opening screen remains open.

overloadColor type: XmRString default: red

Specifies the color of the trace ID displayed when an ADC overload occurs. If no value is set, the trace ID color does not change when an overload condition occurs.

ownColormap type: Boolean default: False

Enables the software to use its own colormap. This is useful when there are not enough colors available in the system colormap. On computers that have a dual hardware colormap system it works well. However, on single hardware colormap systems, going into and out of the E3238S window causes harsh color usage for the window without focus.

You can also use the -ownColormap flag when stating the program from the prompt.

plotColorBackground type: XmRString default: Black

Specifies the color used for the background in the energy history plot window.

plotColorTrace type: XmRString default: White

Specifies the color used for the trace in the energy history plot window.

plotColor1 type: XmRString default: Yellow

Specifies the color used for the marker in the energy history plot window.

plotColor2 type: XmRString default: Green

Specifies the color used for the average value in the energy history plot window.

plotColor3 type: XmRString default: gray60

Specifies the color used for the minimum to maximum range in the energy history plot window.

powerOnSweep type: Boolean default: True

Specifies whether to start sweeping when the software is first started.

powerUpMissionSetup type: XmRString default:

When a filename is specified, the e3238s software's initial state is defined by the mission setup contained in this file.

remoteMode type: Boolean default: False

Prevents the software from displaying an X Window as well as error or message dialog boxes. This allows programs that use E3238S output and need no control via the graphic user interface (e.g., socket port information) to suppress the normal application window. Any unwritten information at exit time is deleted without warning.

See also, hideDisplay (pg 187).

signalDatabaseSize type: Integer default: 500000

Specifies the maximum number of signal database entries allowed. When this maximum number of entries is reached, no new entries are recorded.

spectrogramBackingStoreSize type: Integer default: 100000

Specifies the amount of memory, in bytes, to use for storing the spectrogram and color spectrogram display. This is used when the spectrogram needs to be redisplayed such as when a dialog box is removed from being on top of trace area. This amount of memory is used for each of the four traces. The amount of memory needed for the spectrogram per trace can be computed by:

```
MW = maximum width of the trace (in pixels)
ML = maximum height of the trace (in pixels)

memory required (in bytes) = MW * ML / 8
```

The amount of memory needed for the color spectrogram per trace can be computed by

```
MW = maximum width of the trace (in pixels)
ML = maximum height of the trace (in pixels)

memory required (in bytes) = MW * ML * 4
```

At least 5120 bytes of memory is required for each trace.

spectrogramBackingStoreSizeTraceA type: Integer default: 2000000

spectrogramBackingStoreSizeTraceB type: Integer default: 0

spectrogramBackingStoreSizeTraceC type: Integer default: 0

spectrogramBackingStoreSizeTraceD type: Integer default: 0

This is a companion resource with 'spectrogramBackingStoreSize'. Since the color spectrogram requires a large amount of memory for a full backing store, you may want to allocate individual trace values. When the value specified is zero, the trace uses the `spectrogramBackingStoreSize` value.

syncDisplayEnabled type: Boolean default: True

Enables the software to do a synchronization with the X server at the end of every sweep. This should always be True except when running over a communications link that has a very long message round trip time.

toolbarWrap type: Boolean default: False

Specifies whether the toolbar icons wrap to a new row or truncate at the end of a single row of icons.

To allow sizing of the toolbar pane when toolbarWrap is True, the resource `toolbar.paneMaximum` should be increased to view the maximum number of rows.

tooltips type: Boolean default: True

Specifies whether to display the toolbar tooltips.

traceBackgroundColor type: XmRString default: Black

Specifies the background color for all traces.

traceFont type: XmRString default: 9x15

Specifies the font used for labeling that appears within the trace.

traceGridColor type: XmRString default: Gray50

Specifies the grid color for all traces.

traceLabelColor type: XmRString default: White

Specifies the label color for all traces.

traceLineColor type: XmRString default: cyan3

Specifies the line color for all traces.

traceMarkerColor type: XmRString default: Yellow

Specifies the marker color for all traces.

traceThresholdColor type: XmRString default: Blue

Specifies the threshold color for all traces.

traceTranslations type: XmRString default:

Specifies translations for the traces. A common use of these translations is to map function keys to various command line functions. An action routine, commandLine(), is provided to send strings to the command line of this software. For a list of commands available, see the Command Reference (a PDF file).

Example:

```
*traceTranslations: #override\
 <Key>F2: commandLine("*frequencyFullScale")\n\
 <Key>F3: commandLine("*amplitudeAutoScale")\n\
 <Key>F4: commandLine("*markerMode:on")\n\
 <Key>F5: commandLine("*markerMode:off")\n\
 <Key>F6: commandLine("*frequencyFullScale","*amplitudeAutoScale")\n
```

This example maps five function keys as follows:

- F2 to do a frequency full scale
- F3 to do a amplitude auto scale
- F4 to turn the marker on
- F5 to turn the marker off
- F6 to do both an amplitude and frequency auto scale.

transientTitles type: Boolean default: False

Specifies whether to put titles on transient dialog boxes.

useHardware type: Boolean default: True

Specifies whether or not to use the VXI hardware. When True, energy data is random. You can also start the program with the `-noHardware` flag.

useOldTimeSnapshotFileFormat type: Boolean default: False

The software starting with version C.00.01 implements a new time snapshot file format to allow file sizes larger than 1 GB. To save time snapshot files in the old format, set this value to true. You will not be able to save a file larger than 1 GB.

userColorMapEnabled type: Boolean default: False

Enables user color map code.

Multi-Channel Hardware Installation

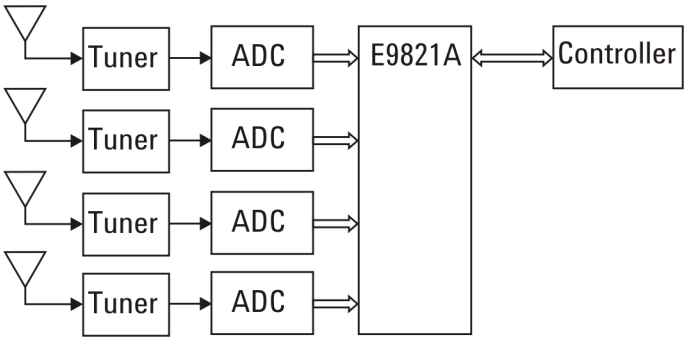
This chapter describes the installation and configuration of measurement hardware for Option EMC, multi-channel search. Option EMC is only available with the E3238S/35688E Signal Intercept and Collection System. Option EMC is not available with the E3238S/N6820E Signal Survey Software.

Overview

This section describes configurations supported by option EMC, Multi-Channel Search. This supports more than one and as many as four search receiver "channels" in an E3238s system.

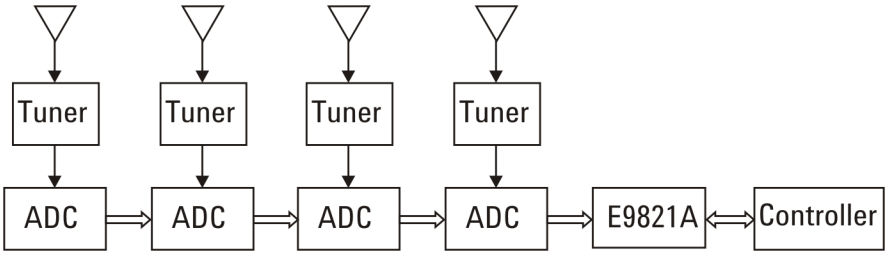
The block diagram in figure 57 illustrates a four-channel configuration. The tuners' IF outputs are digitized by the ADCs. The digitized IF is passed to the signal processor (E9821A) which converts it from time to magnitude data and passes it to the controller. This is, effectively, how the system works. This model effectively describes multi-channel operation.

Figure 57. Block diagram of a multi-channel search system



The N6830A/HF, N6830A/70 MHz IF, and E1439/70 MHz IF configurations incorporate the fiber-optic FPDP (front-panel data port) interface to get data from the ADC's to the E9821A. The E1437 ADC data actually flows through the ADCs to the E9821A as shown in figure 58. For the E1437 HF solution, the data path is the VXI Local Bus so the ADCs must be installed adjacent to each other in the mainframe.

Figure 58. Block diagram showing actual data flow



Time Synchronous Measurements

All channels in a multi-channel search system are inherently time synchronous because of the method used to coordinate the ADC sampling activities.

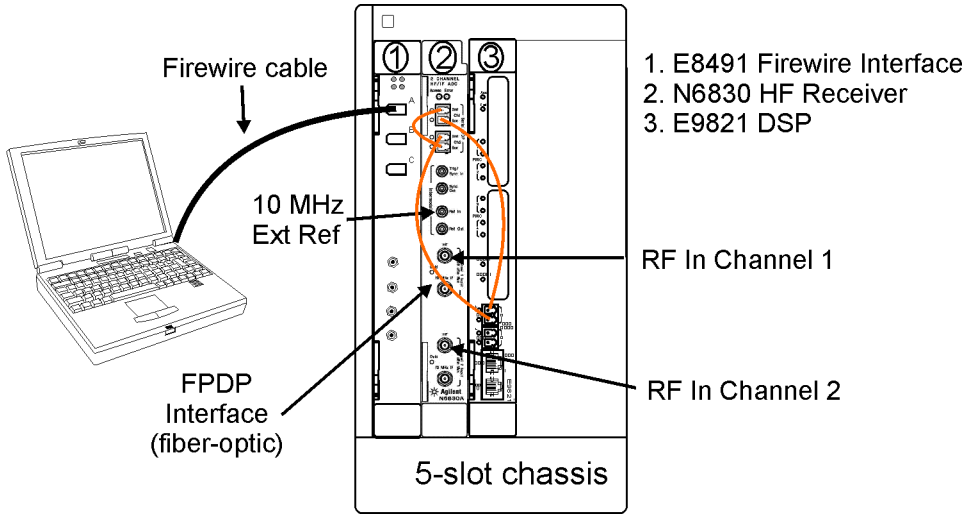
- One ADC is designated as the "master" and it provides a sample-clock signal and a measurement-sync signal to all other ADCs in the system.
- The sample-clock and measurement-sync signals may be passed from the master to the slaves either on the VXI backplane or via the ADC front-panel SMB connectors. The N6830A/HF and N6830A/70 MHz IF use front-panel SMB connectors only.
- The master may run on it's internal frequency reference or an external reference may be provided on the front-panel BNC connector.

HF Configurations

2-channel N6830A HF Configuration

A 2-channel N6830A HF system configuration is shown in figure 59.

Figure 59. N6830A 2 channel HF system



2-channel N6830A HF Configuration Settings

The following is a configuration example for a 2-channel, N6830A HF system for the e3238s.cfg file.

searchRxConfiguration: MultiChannelParallelTimeSynchronousSearch
numSearchRx: 2

searchRx1.dspModel: E9821A
searchRx1.dspModules: 128
searchRx1.dspCmndPort: VXI
searchRx1.dspDataPort: VXI
searchRx1.dspDataCompression: On

searchRx2.dspModel: Shared

searchRxAdcSynchronization: ADCFrontPanel

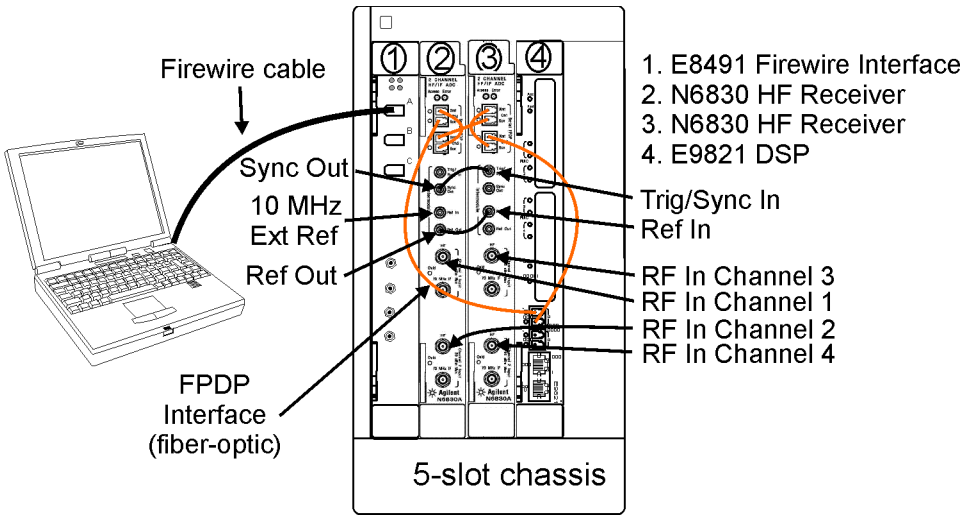
searchRx1.adcModel: N6830A/HF
searchRx1.adcInterfaceParm: 64
searchRx1.adcDataPort: FPDP_NO_CHECK
searchRx1.adcClock: Internal
searchRx1.adcMasterClock: Auto

searchRx2.adcModel: N6830A/HF
searchRx2.adcInterfaceParm: 65
searchRx2.adcDataPort: FPDP_NO_CHECK
searchRx2.adcClock: Internal
searchRx2.adcMasterClock: Auto

4-channel N6830A HF Configuration

A 4-channel N6830A HF system configuration is shown in figure 60.

Figure 60. N6830A 4 channel HF system



4-channel N6830A HF Configuration Settings

The following is a configuration example for a 4-channel, N6830A HF system for the e3238s.cfg file.

searchRxConfiguration: MultiChannelParallelTimeSynchronousSearch
numSearchRx: 4

searchRx1.dspModel: E9821A
searchRx1.dspModules: 128
searchRx1.dspCmndPort: VXI
searchRx1.dspDataPort: VXI
searchRx1.dspDataCompression: On

searchRx2.dspModel: Shared
searchRx3.dspModel: Shared
searchRx4.dspModel: Shared

searchRxAdcSynchronization: ADCFrontPanel

searchRx1.adcModel: N6830A/HF
searchRx1.adcInterfaceParm: 64
searchRx1.adcClock: Internal
searchRx1.adcMasterClock: Auto

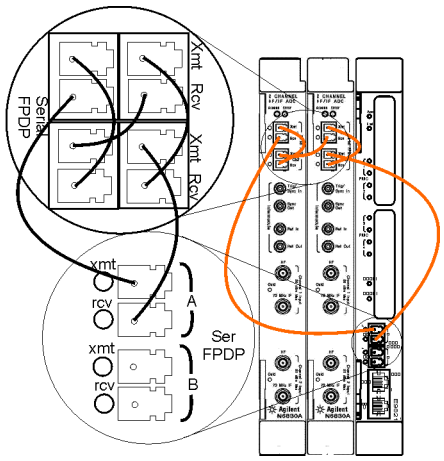
searchRx2.adcModel: N6830A/HF
searchRx2.adcInterfaceParm: 65
searchRx2.adcClock: Internal
searchRx2.adcMasterClock: Auto

searchRx3.adcModel: N6830A/HF
searchRx3.adcInterfaceParm: 66
searchRx3.adcClock: Internal
searchRx3.adcMasterClock: Auto

searchRx4.adcModel: N6830A/HF
searchRx4.adcInterfaceParm: 67
searchRx4.adcClock: Internal
searchRx4.adcMasterClock: Auto

Figure 61 shows a detailed view of the FPDP cabling for a 4 channel N6830A system. The cabling is the same whether you are using the HF or 70 MHz IF inputs on the N6830A.

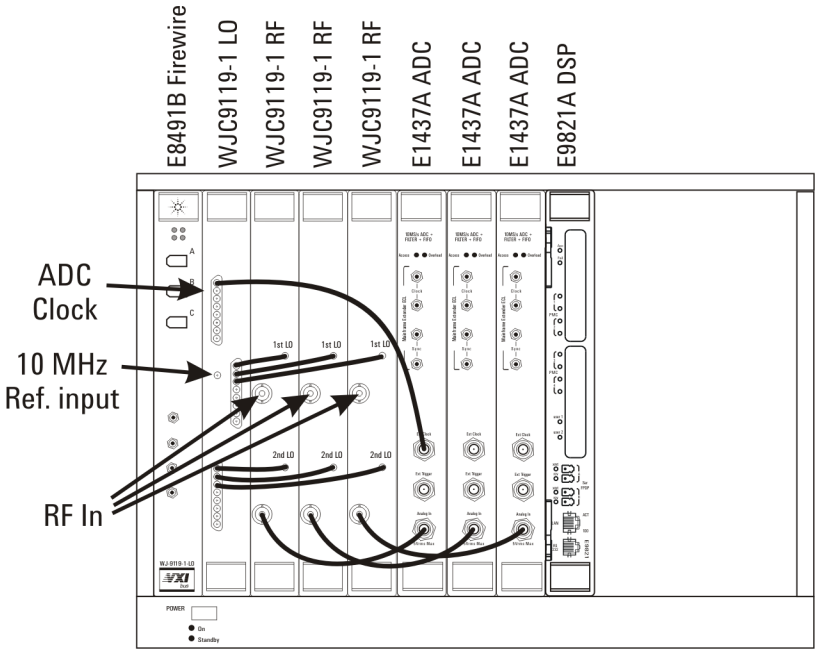
Figure 61.
4 channel N6830A
FPDP cabling



3-channel E1437A HF Configuration

The E1437A HF configuration uses the WJ-9119 LO and downconverter sections and the E1437A ADC modules as shown in figure 62.

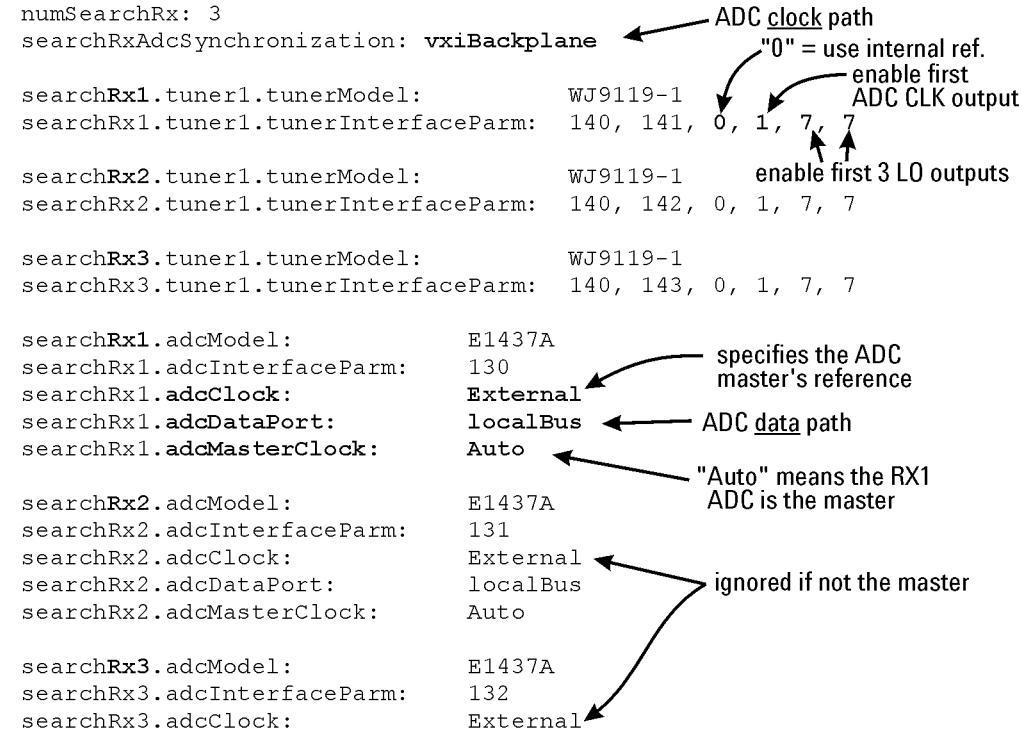
Figure 62. E1437A 3 channel HF system



3-channel E1437A HF Configuration Settings

The following is a configuration example for a 3-channel, E1437 HF system for the e3238s.cfg file.

```
searchRxConfiguration: MultiChannelParallelTimeSynchronousSearch
numSearchRx: 3
searchRxAdcSynchronization: vxiBackplane

searchRx1.tuner1.tunerModel:          WJ9119-1
searchRx1.tuner1.tunerInterfaceParm:  140, 141, 0, 1, 7, 7

searchRx2.tuner1.tunerModel:          WJ9119-1
searchRx2.tuner1.tunerInterfaceParm:  140, 142, 0, 1, 7, 7

searchRx3.tuner1.tunerModel:          WJ9119-1
searchRx3.tuner1.tunerInterfaceParm:  140, 143, 0, 1, 7, 7

searchRx1.adcModel:             E1437A
searchRx1.adcInterfaceParm:     130
searchRx1.adcClock:             External
searchRx1.adcDataPort:          localBus
searchRx1.adcMasterClock:       Auto

searchRx2.adcModel:             E1437A
searchRx2.adcInterfaceParm:     131
searchRx2.adcClock:             External
searchRx2.adcDataPort:          localBus
searchRx2.adcMasterClock:       Auto

searchRx3.adcModel:             E1437A
searchRx3.adcInterfaceParm:     132
searchRx3.adcClock:             External
searchRx3.adcDataPort:          localBus
searchRx3.adcMasterClock:       Auto

searchRx1.dspModel:             E9821A
searchRx1.dspInterfaceParm:     128
searchRx1.dspCmndPort:          VXI
searchRx1.dspDataPort:          VXI
downloadable:                   C:\E3238S\downloadables\e9821a.out

searchRx2.dspModel:             Shared
searchRx3.dspModel:             Shared
```



General Configuration

`searchRxConfiguration:` `MultiChannelParallelTimeSynchronousSearch` specifies the multi-channel search mode; all channels sweep the same frequency range.

`numSearchRx` specifies the number of search receiver channels installed.

`searchRxAdcSynchronization` specifies whether the ADC clock and Sync signals are passed from one ADC to the next via the VXI backplane or the ADC front panel connectors. The default is `vxiBackplane`.

Tuner Configuration

Each channel (searchRx1, searchRx2, etc.) has a set of tuner parameter settings. There is one LO module per system and one downconverter (RF) module for each channel. The LO parameters are defined by the `tunerInterfaceParm` as follows:

- The first two parameters are the **logical address** for the LO and RF modules. The last four parameters are LO settings which are defined in the Rx1 definition as shown in the listing (they are repeated for each channel).
- The third parameter enables using an **external reference** attached to the LO. Using an external reference increases the absolute frequency accuracy of the system. The default configuration is to use the LO's internal reference.
- The fourth parameter enables the **ADC clock output**. This should be set to 1.
- The fifth and sixth parameters define which 1st and 2nd **LO outputs** are enabled:

  "1" activates the first LO output
  "3" activates the first two LO outputs
  "7" activates LO outputs 1, 2, and 3

ADC Configuration

The first two parameters listed are the model and logical address.

ADC Clock Background

Each ADC is either a master or a slave with respect to the "shared" clock. There is only one master in a system.

- For the designated master:
  - The clock source is specified by the `adcClock` parameter; either internal or external (front-panel BNC connector labeled "Ext Clock")
  - The `searchRxAdcSynchronization` parameter specifies where to send the clock for the slaves to use; either the VXI backplane or the ADC front-panel SMB connectors.
- All slaves use the clock provided by the master
  - The clock source is specified by the `searchRxAdcSynchronization` parameter
  - The `adcClock:` parameter is not meaningful for slaved ADCs

`adcClock` specifies the clock source:

- **Internal** means it uses a clock signal generated in the module.
- **External** means that, if this is the clock master, the ADC takes its clock from the External Clock front-panel connector.

`adcMasterClock` specifies whether the ADC provides its sample clock for other devices (which makes it the master). Whether this clock is provided on the front panel or on the VXI backplane is specified with `searchRxAdcSynchronization`.

This parameter is either On, Off, or Auto.

- **On** means it provides the ADC clock (it is the master).
- **Off** means that it does not provide its ADC clock.
- **Auto** means that master/slave status is set automatically. This is the recommended setting.
  - The ADC designated as searchRx1 is the clock master
  - Other (slave) ADCs get their clock as specified by `searchRxAdcSynchronization`

`adcDataPort` specifies the data path from the ADC to the DSP. The E1437A does not support the fiber-optic FPDP so `localBus` is the only acceptable value.

DSP Configuration

The first two parameters listed are the model and logical address.

`dspCmndPort` specifies the path used by the controller to pass commands to the DSP. As of this writing, VXI is the only valid setting.

`dspDataPort` specifies the path used to pass DSP output data to the controller. As of this writing, VXI is the only valid setting.

`downloadable` specifies the file to be downloaded from the controller to the DSP.

DSP Hardware Optimization

The E9821A is a carrier for ePMC cards. There are mounting locations for four cards on the E9821A. Option 101 is a card containing two G4 processors. The E9821A supports as many as three of these[1].

The optimum number of cards depends on whether the sweep settings result in performance that is bound by DSP resources or the IO throughput between the DSP and controller.

Synchronization

The WJ9119-1 HF tuner has a separate LO which drives the downconverter modules and provides the ADC sample clock.

The relative frequency accuracy is optimized by the fact that one LO drives all the channel downconverters.

To optimize the absolute frequency accuracy, just apply a 10 MHz reference signal to the LO 10 MHz REF IN connector.

Time synchronization is optimized by the manner in which the ADC sample clock and sync signals are shared, either on the backplane or on the ADC front-panel connectors. See pg 194.

Phase synchronization requires a calibration algorithm that is not currently available.

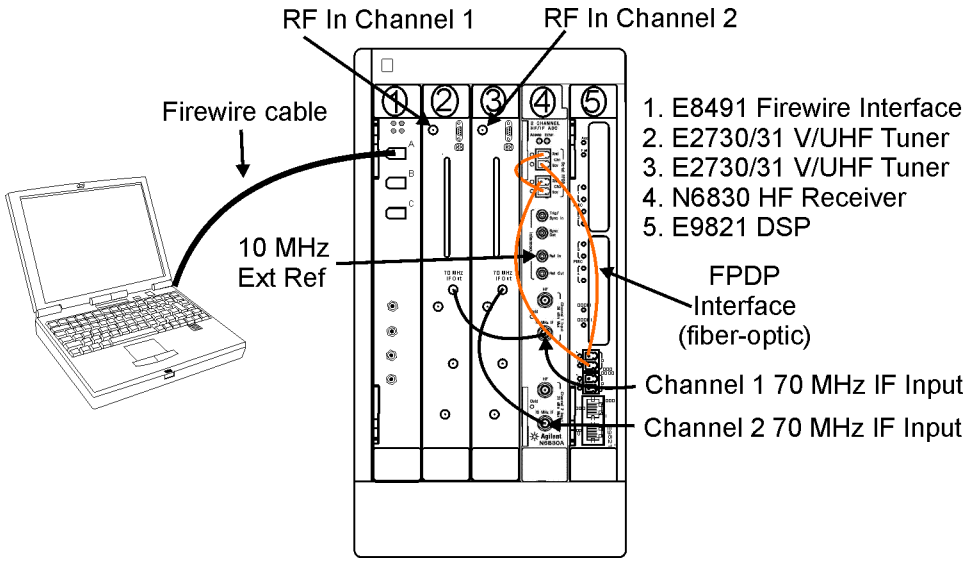
[1]This is true for modules with a serial number above US43140000. Modules with serial numbers below that can support as many as two option-101 boards.

VHF/UHF Configurations

2-channel N6830A V/UHF Configuration

Figure 63 shows a 2-channel VHF/UHF system using the E2730/31 tuners and the N6830A ADCs.

Figure 63. VHF/UHF 2-channel N6830A system



2-channel N6830A V/UHF Configuration Settings

The following is a configuration example for a 2-channel, N6830A VHF/UHF system for the e3238s.cfg file.

searchRxConfiguration: MultiChannelParallelTimeSynchronousSearch
numSearchRx: 2
searchRxAdcSynchronization: ADCFrontPanel

searchRx1.tuner1.tunerModel: E2731B
searchRx1.tuner1.tunerInterface Parm: 136, 0, 3, 20

searchRx2.tuner1.tunerModel: E2731B
searchRx2.tuner1.tunerInterface Parm: 137, 0, 3, 20

searchRx1.dspModel: E9821A
searchRx1.dspModules: 128
searchRx1.dspCmndPort: VXI
searchRx1.dspDataPort: VXI
searchRx1.dspDataCompression: On

searchRx2.dspModel: Shared

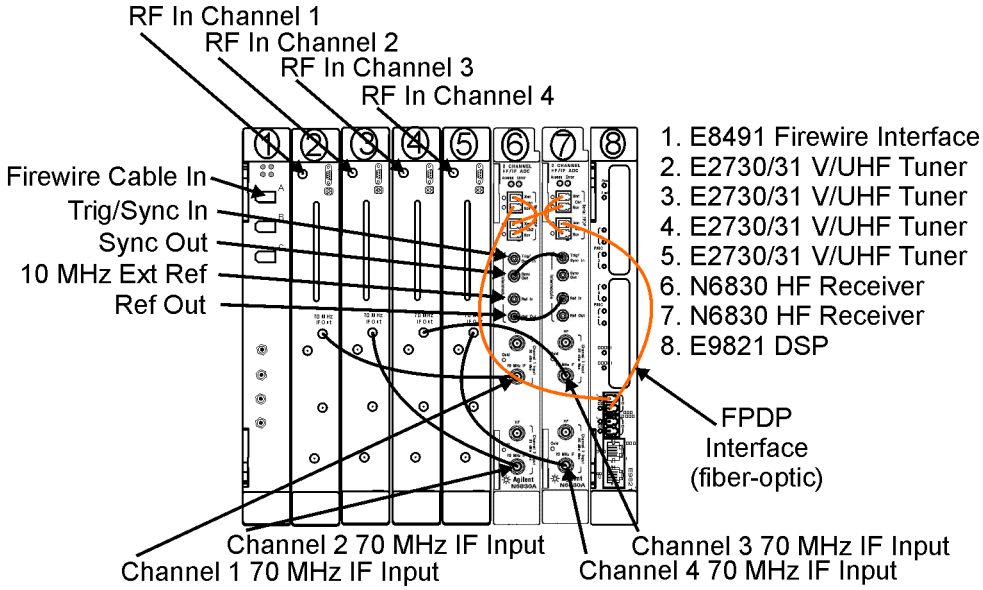
searchRx1.adcModel: N6830A/70
searchRx1.adcInterfaceParm: 64
searchRx1.adcClock: Internal
searchRx1.adcMasterClock: Auto

searchRx2.adcModel: N6830A/70
searchRx2.adcInterfaceParm: 65
searchRx2.adcClock: Internal
searchRx2.adcMasterClock: Auto

4-channel N6830A V/UHF Configuration

Figure 64 shows a 4-channel VHF/UHF system using the E2730/31 tuners and the N6830A ADCs.

Figure 64. VHF/UHF 4-channel N6830A system



2-channel N6830A V/UHF Configuration Settings

The following is a configuration example for a 4-channel, N6830A VHF/UHF system for the e3238s.cfg file.

searchRxConfiguration: MultiChannelParallelTimeSynchronousSearch
numSearchRx: 4
searchRxAdcSynchronization: ADCFrontPanel

searchRx1.tuner1.tunerModel: E2731A
searchRx1.tuner1.tunerInterface Parm: 136, 0, 3, 20

searchRx1.tuner2.tunerModel: E2731A
searchRx1.tuner2.tunerInterface Parm: 137, 0, 3, 20

searchRx1.tuner3.tunerModel: E2731A
searchRx1.tuner3.tunerInterface Parm: 138, 0, 3, 20

searchRx1.tuner4.tunerModel: E2731A
searchRx1.tuner4.tunerInterface Parm: 139, 0, 3, 20

searchRx1.dspModel: E9821A
searchRx1.dspModules: 128
searchRx1.dspCmndPort: VXI
searchRx1.dspDataPort: VXI
searchRx1.dspDataCompression: On

searchRx2.dspModel: Shared
searchRx3.dspModel: Shared
searchRx4.dspModel: Shared

searchRx1.adcModel: N6830A/70
searchRx1.adcInterfaceParm: 64
searchRx1.adcClock: Internal
searchRx1.adcMasterClock: Auto

searchRx2.adcModel: N6830A/70
searchRx2.adcInterfaceParm: 65
searchRx2.adcClock: Internal
searchRx2.adcMasterClock: Auto

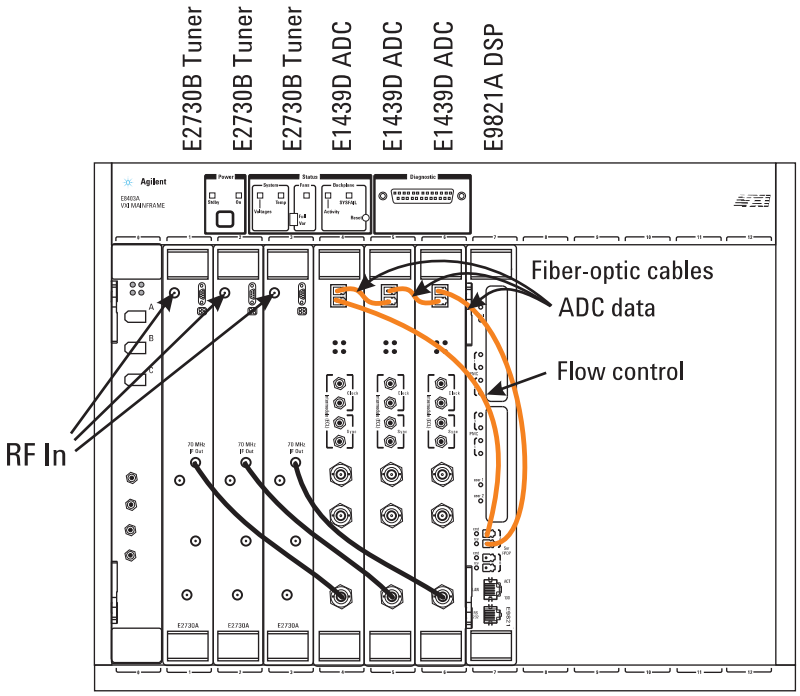
searchRx3.adcModel: N6830A/70
searchRx3.adcInterfaceParm: 66
searchRx3.adcClock: Internal
searchRx3.adcMasterClock: Auto

searchRx4.adcModel: N6830A/70
searchRx4.adcInterfaceParm: 67
searchRx4.adcClock: Internal
searchRx4.adcMasterClock: Auto

3-channel E1439D V/UHF Configuration

This VHF/UHF configuration uses the E2730/31 tuners and E1439A/B/D ADC modules as shown in figure 65

Figure 65. VHF/UHF 3-channel E1439D system



3-channel E1439D V/UHF Configuration Settings

This configuration is defined in the following configuration-file listing:

Figure 66. Configuration settings VHF/UHF

```
searchRxConfiguration: MultiChannelParallelTimeSynchronousSearch
numSearchRx: 3
searchRxAdcSynchronization: vxiBackplane

searchRx1.tuner1.tunerModel:          E2730A
searchRx1.tuner1.tunerInterfaceParm:  136, 0, 3, 20

searchRx2.tuner1.tunerModel:          E2730A
searchRx2.tuner1.tunerInterfaceParm:  137, 0, 3, 20

searchRx3.tuner1.tunerModel:          E2730A
searchRx3.tuner1.tunerInterfaceParm:  138, 0, 3, 20

searchRx1.adcModel:            E1439B/70
searchRx1.adcInterfaceParm:    130
searchRx1.adcClock:            Internal
searchRx1.adcDataPort:         FPDP
searchRx1.adcMasterClock:      Auto

searchRx2.adcModel:            E1439B/70
searchRx2.adcInterfaceParm:    131
searchRx2.adcClock:            Internal
searchRx2.adcDataPort:         FPDP
searchRx2.adcMasterClock:      Auto

searchRx3.adcModel:            E1439B/70
searchRx3.adcInterfaceParm:    132
searchRx3.adcClock:            Internal
searchRx3.adcDataPort:         FPDP
searchRx3.adcMasterClock:      Auto

searchRx1.dspModel:            E9821A
searchRx1.dspInterfaceParm:    128
searchRx1.dspCmndPort:         VXI
searchRx1.dspDataPort:         VXI
downloadable:                  C:\E3238S\downloadables\e9821a.out

searchRx2.dspModel:            Shared
searchRx3.dspModel:            Shared
```

General Configuration

`searchRxConfiguration:` `MultiChannelParallelTimeSynchronousSearch` specifies the multi-channel search mode; all channels sweep the same frequency range.

`numSearchRx` specifies the number of search receiver channels installed.

`searchRxAdcSynchronization` specifies whether the ADC clock and Sync signals are passed from one ADC to the next via the VXI backplane or the ADC front panel connectors. The default is `vxiBackplane`.

Tuner Configuration

Each channel (searchRx1, searchRx2, etc.) has a set of tuner parameter settings:

- The first parameter is the **logical address** for the downconverter module.
- The second parameter specifies the **10 MHz reference** source.
  - 0 selects the internal reference. This is the default setting.
  - 1 selects the external reference connection on the tuner's front panel.
  - 2 selects the external reference on the VXI backplane. This is from the controller and requires configuring the controller to provide it properly.
- The third parameter specifies the tuner's **settling time**.
- The fourth parameter specifies the **lowest frequency** to which the downconverter can tune (MHz). The default is 20 MHz but the lowest possible setting is 2 MHz. Note: The E2730 model does not perform well in the HF region. There are significant spurs and distortion components below 20 MHz.

ADC Configuration

The first two parameters listed are the model and logical address.

ADC Clock Background

Each ADC is either a master or a slave with respect to the "shared" clock. There is only one master in a system.

- For the designated master:
  - The clock source is specified by the `adcClock` parameter; either internal or external (front-panel BNC connector labeled "Ext Clock")
  - The `searchRxAdcSynchronization` parameter specifies where to send the clock for the slaves to use; either the VXI backplane or the ADC front-panel SMB connectors.
- All slaves use the clock provided by the master
  - The clock source is specified by the `searchRxAdcSynchronization` parameter
  - The `adcClock:` parameter is not meaningful for slaved ADCs

`adcClock` specifies the clock source:

- **Internal** means it uses a clock signal generated in the module.
- **External** means that, if this is the clock master, the ADC takes its clock from the External Clock front-panel connector.

`adcMasterClock` specifies whether the ADC provides its sample clock for other devices (which makes it the master). Whether this clock is provided on the front panel or on the VXI backplane is specified with `searchRxAdcSynchronization`.

This parameter is either On, Off, or Auto.

- **On** means it provides the ADC clock (it is the master).
- **Off** means that it does not provide its ADC clock.
- **Auto** means that master/slave status is set automatically. This is the default.
  - The ADC designated as searchRx1 is the clock master
  - The other (slave) ADCs get their clock as specified by `searchRxAdcSynchronization`

`adcDataPort` specifies the path for the ADC data from the ADC to the E9821 DSP module. This is shipped set to `localBus` so that it works without the fiber-optic cables. For optimal performance change this to `FPDP` (front panel data port) and connect the orange fiber-optic lines as shown in figure 65 on page 180.

DSP Configuration

The first two parameters listed are the model and logical address.

`dspCmndPort` specifies the path used by the controller to pass commands to the DSP. As of this writing, VXI is the only valid setting.

`dspDataPort` specifies the path used to pass DSP output data to the controller. As of this writing, VXI is the only valid setting.

`downloadable` specifies the file to be downloaded from the controller to the DSP.

DSP Hardware Optimization

The E9821A is a carrier for ePMC cards. There are mounting locations for four cards on the E9821A. Option 101 is a card containing two G4 processors. The E9821A supports as many as three of these[1].

The optimum number of cards depends on whether the sweep settings result in performance that is bound by DSP resources or the IO throughput between the DSP and controller.

Synchronization

The E2730/31/A/B tuners have the ability to share their LO outputs with one other tuner but, as of this writing, it has not been satisfactorily tested. The best that can be done to synchronize multiple channels is to provide a common 10 MHz reference to all tuners. The controller can be configured to provide this via the VXI backplane but you should expect degraded performance in the form of increased LO sideband levels.

Note

If the tuners do not have a common 10 MHz reference then there will be relative frequency errors (frequency accuracy between channels) that may be significant for some applications.

Time synchronization is optimized by the manner in which the ADC sample clock and sync signals are shared, either on the backplane or on the ADC front-panel connectors. See pg 161.

Phase synchronization requires a calibration algorithm that is not currently available.

[1]This is true for modules with a serial number above US43140000. Modules with serial numbers below that can support as many as two option-101 boards.

Specifications

Note Specifications are subject to change without notice.

Note For N6841A RF Sensor specifications, refer to the documentation that came with the N6841A RF Sensor.

Definitions

Specifications Specifications describe the performance of parameters covered by the product warranty

Characteristics Characteristics describe the product performance that is useful in the application of the product, but is not covered by the product warranty.

Typical Typical indicates performance within specification that 80% of the units exhibit with a 80% confidence level over the temperature range 20 to 30 °C. Typical specifications are not covered by product warranty.

Nominal Nominal values indicate the level that all units are expected to surpass over the specified environmental operating range. Nominal specifications are not covered by product warranty.

Supplemental Information Supplemental information encompasses characteristics, typical, nominal, and other information pertinent to understanding the expected, but unwarranted, performance of a given parameter.

Recommended Calibration Interval The recommended calibration interval is the period after its last calibration during which a product is expected to meet its specified parameters. The recommended calibration interval is expressed in months or years.

Calibration Calibration, as Agilent defines it, is the process of verifying that the product meets its warranted specifications and performing adjustments as necessary to either correct any out-of-tolerance conditions or better optimize a parameter to improve the probability that the parameter will be in-tolerance at the calibration.

Room Temperature Room temperature is generally accepted to be the range from 20 to 30 °C.

Conditions, Certification, and Calibration

Conditions Required to Meet Specifications.

All conditions must be met.

- The product is being operated within the specified conditions for temperature, altitude, and humidity
- Any system components that specify a calibration cycle must calibrated
- Spectrum Corrections must be enabled in the E3238S software application
- The product has been warmed up for at least 30 minutes.

Certification

Agilent Technologies certifies that this product met its published specifications at the time of shipment from the factory. Agilent Technologies further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology, to the extent allowed by the Institute's calibration facility, and to the calibration facilities of other International Standards Organization members.

Calibration Cycle

Calibration requirements are specific to each component of the system. Components such as the VXI mainframe and E9821 Signal Processor VXI module do not require periodic calibration. Other VXI modules and instruments, such as tuners and ADCs, may recommend periodic calibration or performance verification procedures. Refer to the documentation for each hardware component.

E3238S Hardware Configurations

The specifications and supplemental information listed in this document apply to the following hardware configurations.

E3238S Configuration Name	Frequency Range	RF Downconverter and Tuner	ADC module	Signal Processing Module	Document Section
N6830 HF System	100 kHz to 32 MHz	No downconverter required. The N6830 is an HF receiver	N6830	E9821A	Section Specifications for N6830 100 kHz to 32 MHz VXI System
2.7 GHz VXI	20 MHz to 2.7 GHz	E2730B VXI module	E1439D or N6830A	E9821A	Section Specifications for 2.7 GHz VXI System and 6 GHz VXI System
6 GHz VXI	20 MHz to 6.0 GHz	E2731B VXI module	E1439D or N6830A	E9821A	Section Specifications for 2.7 GHz VXI System and 6 GHz VXI System
26 GHz PSA System	100 kHz to 26.5 GHz	E4440A PSA Spectrum Analyzer with Option HY7[1]	E1439D	E9821A	Section Specifications for 26 GHz PSA Systems

[1]Option HY7 adds a 70 MHz IF Output to the PSA, which is required for use in E3238S systems. Option H70 is also supported but HY7 is preferred because of it improves residual response performance.

The following table lists additional hardware E3238S configurations. These configurations are not covered by this document. Contact Agilent for information on these configurations.

E3238S Configuration Name	Frequency Range	RF Downconverter and Tuner	ADC module	Signal Processing Module
Other PSA Systems	to 6.7 GHz to 13.2 GHz	E4443A PSA + HY7, E4445A PSA + HY7	E1439D	E9821A
18 GHz VXI System	20 MHz to 18 GHz	VME downconverter module, and E2730B VXI module	E1439D	E9821A
E1437A 32 MHz VXI HF System	100 kHz to 32 MHz	VXI HF tuner module	E1437A	E9821A

Tuner Sweep Control (Locked Tuner mode)

“Tuner Lock” is a setting of the application software for this system. It appears as a check box under Search, Type, Tuner Sweep Control. The Tuner Lock setting affects specifications such as frequency span and affects other system capabilities.

When Tuner Lock is Off, the system can search over a large frequency range by rapidly stepping the tuner center frequency. At each frequency step, a span as wide as 36 MHz is measured.

When Tuner Lock is On, the tuner is held at a fixed frequency, and the system “stares” at a portion of the spectrum (up to 36 MHz wide). This enables continuous data to be sent to narrowband signal processing algorithms. These options include Multi-channel Narrow-band Recorder (NBR) and Audio Output (AU1)

The following table summarizes the two modes.

Feature	Tuner Lock Off	Tuner Lock On
Search Frequency Span	Full frequency range of tuner	Limited to 36 MHz
Narrowband Signal Processing, such as Narrow-band Recorder or Audio Output	Not available	Available
Time Snapshot captures	Available (between sweeps)	Not Available
Option MR1, Host-based Modulation Recognition Option	Available (between sweeps)	Available only when used with option USD, (Universal Signal Detection)
Min Resolution Bandwidth (RBW)	8.5 Hz	140 Hz

Specifications for N6830 100 kHz to 32 MHz VXI System

This section lists the specifications and supplemental information that applies to the hardware configurations that use the HF inputs of the N6830A dual channel HF receiver and 70 MHz IF ADC. This measurement hardware consists of a VXI mainframe and three VXI modules: one E8491 firewire module, one N6830 dual channel HF receiver, and one E9821A Signal Processing module.

Frequency

Description	N6830A/HF Nominal Specification
Frequency Range	100 kHz to 32 MHz
Frequency Reference	The N6830 module can be configured to use its internal reference or an external 10 MHz reference. Frequency accuracy when using the internal reference is ±10 ppm from 0 to 50 deg C
Frequency Span	
Tuner unlocked	100 kHz to 32 MHz
Time Snapshot File Frequency Spans.	500 Hz to 32 MHz in 2x steps[1]
Maximum Tuner Locked Search Stare Bandwidth	32 MHz
Tuner locked, Narrowband processing bandwidth, each channel	600 Hz to 102 kHz
Resolution Bandwidth (RBW)	
RBW filters and shape factors	Hanning 9 : 1, Gausstop 4 : 1, Flattop 2.6 : 1
RBW range (for 9:1 shape factor)	15 Hz to 240 kHz in 2x steps[2]
Number of frequency points maintained in spectrum display results (at minimum RBW)	176,600 points per MHz of frequency span. Available PC memory may limit the number of points[3].
Phase Specifications	
Phase Noise Density (single sideband power density, absolute or residual. <0.05 G vibration)	
100 kHz Offset	< -140 dBc/Hz (typical)
10 kHz Offset	< -140 dBc/Hz (typical)
1 kHz Offset	< -125 dBc/Hz (typical)
100 Hz Offset	< -105 dBc/Hz (typical)

[1]The maximum frequency span depends on the chosen ADC sample rate. Set the sample rate in the searchRx1.adcSampleRate: line of the e3238s.cfg file. 20480000 = 8 MHz, 40960000 = 16 MHz, 81920000 = 32 MHz.

[2]When running narrowband signal processing such as Narrow-band Recorder, the minimum RBW is 60 Hz. The maximum RBW is 240 kHz

[3]Each frequency point requires 4 bytes of host computer memory.

Amplitude

Description	N6830A/HF Nominal Specification
Amplitude Range	
Full-scale level (ADC overload level) at RF input (Preset, 0 dB Attenuation)	-12 dBm
Input Attenuation	-12 dB to +18 dB in 2 dB steps
Additional Tuner Attenuation	0 dB (Attenuator Disabled) or 18 dB (Attenuator Enabled)
Amplitude Accuracy	
Direct, No preselection filters	±3.2 dB
High Pass/Low pass preselection filters, except 2 MHz High Pass	±3.5 dB
2 MHz High Pass preselection filter	±5.8 dB
Noise and Sensitivity	
Noise level	
0 dB attenuation (preset)	-143 dBfs/Hz
Noise figure (measured on lowest range)	13 dB
Dynamic Range	
Intermodulation Distortion Two in-band signals 500 kHz apart, ≤ -8 dBfs	
Third Order Intercept (TOI) at -12dBm range at -8 dBfs	15 dBm (nominal) 27 dBm (typical)
Third Order Distortion	< -80 dBc
Residual spurs (≥ 3MHz)	-118 dBm (nominal) -121 dBm (typical)
Residual spurs (< 3MHz)	-75 dBm (nominal) -84 dBm (typical)
Sideband Spurs	< -110 dBc
Crosstalk	< -80 dB
Harmonic Distortion	< -80 dBc or -100 dBfs
Aliased Harmonic Distortion	< -70 dBc

Inputs and Outputs

Description	N6830A/HF Nominal Specification
RF input	
Connector	SMA female, 50Ω
VSWR at tuned frequency	< 2.0 : 1 (typical)
Pre-selection filtering[1]	
High Pass Filter	0.1, 2, 5, 9, 15, and 22 MHz
Low Pass Filter	5,9,15,22, and 32 MHz
Absolute Maximum Input Power (Damage Level)	+30 dBm

[1]Pre-select filters will be chosen automatically dependent on the start and stop frequencies of the sweep. In directed sweep mode, the start frequency is the lowest start frequency used in any active band, and the stop frequency is the highest stop frequency used in any active band.

Software Features and Benchmarks

Description	Nominal Specification
E3238S Software Features	See the E3238S Options list at, http://www.agilent.com/find/n6820e.

Specifications for 2.7 GHz VXI System and 6 GHz VXI System

This section lists the specifications and supplemental information that applies to the hardware configurations that use the 2.7 GHz or 6 GHz VXI tuner. This measurement hardware consists of a VXI mainframe and four VXI modules: one E8491 firewire module, one E2730B or E2731B tuner module, one N6830A or E1439D ADC module, and one E9821A Signal Processing module.

Frequency

Description	E273XB/E1439D Nominal Specification	E273XB/N6830A Nominal Specification
Frequency Range		
E2730B VXI tuner	20 MHz to 2.7 GHz[1]	20 MHz to 2.7 GHz[1]
E2731B VXI tuner	20 MHz to 6.0 GHz[1]	20 MHz to 6.0 GHz[1]
Frequency Reference	The tuner module and ADC module can be configured to use their internal references or an external 10 MHz reference. Frequency accuracy when using the internal reference is ±7 ppm.	The tuner module and N6830A can be configured to use their internal references or an external 10 MHz reference. N6830A Internal reference frequency accuracy is ±10 ppm from 0 to 50 deg C
Frequency Span		
Tuner unlocked	20 MHz to 2.7 GHz (or to 6 GHz)	20 MHz to 2.7 GHz (or to 6 GHz)
Time Snapshot File Frequency Spans.	600 Hz to 37.1 MHz in 2x steps	600 Hz to 37.1 MHz in 2x steps
Maximum Tuner Locked Search Stare Bandwidth	36 MHz	36 MHz
Tuner locked, Narrowband processing bandwidth, each channel[2]	1.9 kHz to 340 kHz	1.9 kHz to 340 kHz
Resolution Bandwidth (RBW)		
RBW filters and shape factors	Hanning 9 : 1, Gausstop 4 : 1, Flattop 2.6 : 1	Hanning 9 : 1, Gausstop 4 : 1, Flattop 2.6 : 1
RBW range (for 9:1 shape factor)	8.5 Hz to 1.11 MHz in 2x steps[3]	8.5 Hz to 1.11 MHz in 2x steps[3]
Number of frequency points maintained in spectrum display results (at minimum RBW)	176,600 points per MHz of frequency span. Available PC memory may limit the number of points[4].	176,600 points per MHz of frequency span. Available PC memory may limit the number of points[4].
Phase Specifications		
Phase Noise (Noise at various frequency offsets from a 1 GHz CW signal, tuner locked)		
20 kHz Offset	< -99 dBc/Hz[5]	< -99 dBc/Hz[5]
100 kHz Offset	< -110 dBc/Hz[5]	< -110 dBc/Hz[5]

[1]Tuning from 2 MHz to 20 MHz is possible but may reduce dynamic range.

[2]With ADC sample rate set to 95 MSamples/Sec

[3] When running narrowband signal processing such as Narrow-band Recorder, the minimum RBW is 140 Hz. The maximum RBW is 556.65 kHz

[4]Each frequency point requires 4 bytes of host computer memory.

[5]Limited by the E2730/31 tuner.

Amplitude

Description	E273XB/E1439D Nominal Specification	E273XB/N6830A Nominal Specification
Amplitude Range		
Full-scale level (ADC overload level) at RF input (Preset, 0 dB Attenuation)	– 20 dBm	– 20 dBm
Input Attenuation	0 to 56 dB attenuation in 2 dB steps[1]	0 to 56 dB attenuation in 2 dB steps
IF gain control via ADC range	– 12 dB to +36 dB gain, relative to 0 dBm preset value, in 3 dB steps	– 8 dB to +22 dB gain, relative to 0 dBm preset value, in 2 dB steps
Amplitude Accuracy (Corrections enabled)		
Absolute Accuracy at 50 MHz	±1 dB	±3 dB
Amplitude response flatness		
<100 MHz span below 2.7 GHz	< 1.5 dB peak-to-peak	< 1.5 dB peak-to-peak (typical)
Full span	< 3.0 dB peak-to-peak	< 3.0 dB peak-to-peak (typical)
Noise and Sensitivity		
Noise level (1 GHz).		
ADC range 0 dBm (preset)	– 146 dBm/Hz	< – 160 dBm/Hz
ADC range < – 15 dBm	– 158 dBm/Hz, equivalent to 16 dB Noise Figure	< – 160 dBm/Hz
Dynamic Range		
3rd order intermodulation distortion and input-related spurious responses. (– 30 dBm input signal, excluding harmonics of RF frequencies).	<– 70 dBc	-85 dBc or -105 dBfs
Third order intercept (TOI) of RF input		
< 2.7 GHz	> +5 dBm	> +5 dBm[2]
> 2.7 GHz	> +1 dBm	> +1 dBm[2]
Residual spurs	<– 110 dBm	<– 115 dBm

[1] This electronic attenuator is located after the first mixer of the tuner. To maintain dynamic range, its setting should not exceed about 10 dB. In the E3238S software, this attenuator setting can be different for each band of a Directed Search.

[2]Limited by the E2730/31 tuner

Inputs and Outputs

Description	E273XB/E1439D Nominal Specification	E273XB/N6830A Nominal Specification
RF input		
Connector	SMA female, 50Ω	SMA female, 50Ω
VSWR at tuned frequency	< 2.0 : 1	< 2.0 : 1 (typical)
Pre-selection filtering	none	none
LO level at RF input		
< 2.7 GHz	< – 90 dBm (typical)	< – 90 dBm (typical)
> 2.7 GHz	< – 80 dBm (typical)	< – 80 dBm (typical)
Absolute Maximum Input Power (Damage Level)	+20 dBm	+20 dBm

Shielding Effectiveness

Description	Typical Specification
Chassis Shielding Effectiveness Level of displayed signal due to ambient E-Field with no RF input.	73 dB/m Applied RF field: 1 V/m per IEC 61326-1 and IEC 61000-4-3 Displayed signal level: -60 dBm (-73 dBV)
Power Supply Shielding Effectiveness Level of displayed signal due to RF energy on power line with no RF input	73 dB Injected RF energy: 1V on power line per IEC 61326-1 and IEC 61000-4-6 Displayed signal level: -60 dBm (-73 dBV)

Software Features and Benchmarks

Description	Nominal Specification
E3238S Software Features	See the E3238S Options tab at, http://www.agilent.com/find/n6820e.

Specifications for 26 GHz PSA Systems

This section lists the specifications and supplemental information that applies to the hardware configurations that use the E4440A PSA Spectrum Analyzer with VXI ADC and signal processing modules. The PSA is used as a tuner, and it must have option HY7, a 70 MHz IF output. This 70 MHz IF signal is connected to the E1439D ADC module in a VXI mainframe which also contains a E9821A signal processing module.

Frequency

Description	Nominal Specification
Frequency Range	
E4440A PSA option HY7	100 kHz to 26.5 GHz
Other PSA models	Varies with model
Frequency Reference	±0.2 ppm
Frequency Span	
Tuner unlocked	199 kHz to 26.5 GHz, depends on PSA model
Tuner locked Frequency Spans, and Snapshot File Frequency Spans.	600 Hz to 37.1 MHz in 2x steps
Tuner locked, narrowband processing bandwidth, each channel	1.9 kHz to 340 kHz
Resolution Bandwidth (RBW)	
RBW filters and shape factors	Hanning 9 : 1, Gausstop 4 : 1, Flattop 2.6 : 1
RBW range (for 9:1 shape factor)	8.5 Hz to 1.11 MHz in 2x steps[1]
Number of frequency points maintained in spectrum display results (at minimum RBW)	176,600 points per MHz of frequency span. Available PC memory may limit the number of points.[2]
Frequency stability	
Phase Noise (Noise at various frequency offsets from a 1 GHz CW signal, tuner locked)	
20 kHz Offset	< -99 dBc/Hz
100 kHz Offset	< -110 dBc/Hz

[1] When running narrowband signal processing such as Narrowband Recorder, the minimum RBW is 140 Hz.

[2]Each frequency point requires 4 bytes of host computer memory.

Amplitude

Description	Nominal Specification
Amplitude Range	
Full-scale level (ADC overload level) at RF input (Preset, 0 dB Attenuation)	– 20 dBm
Preamp off	– 18 dBm
Option 1DS Preamp on[1]	– 46 dBm
Input Attenuator	0 to 50 dB attenuation in 2 dB steps
IF gain control via ADC range	– 36 dB to +12 dB gain, relative to preset value, in 3 dB steps
Amplitude Accuracy	
Absolute Accuracy at 50 MHz	±2.5 dB[2]
Amplitude response flatness	
< 3 GHz	< 2.0 dB peak-to-peak
> 3 GHz	< 6.0 dB peak-to-peak
Noise and Sensitivity	
Noise level (1 GHz).	
Preamp off	– 146 dBm/Hz
Option 1DS Preamp on	– 167 dBm/Hz, equivalent to 7 dB Noise Figure
Dynamic Range	
3rd order intermodulation distortion and input-related spurious responses. (– 28 dBm input signal, excluding harmonics of RF frequencies).	<– 70 dBc
Third order intercept (TOI) of RF input	>+16 dBm
Residual responses, Option HY7	<– 100 dBm
Residual responses, Option H70	<– 94 dBm

[1] This preamp is controlled by a mechanical switch which cannot be changed by the E3238S software. Use of the preamp limits the frequency range to < 3 GHz.

[2] Absolute Accuracy uncertainty can be improved to ±1 dB by measuring the conversion gain of the PSA Option HY7 and entering the value in a configuration file.

Inputs and Outputs

Description	Nominal Specification
RF input	
Connector	SMA female, 50Ω
VSWR at tuned frequency	< 2.0 : 1
Pre-selection filtering	none
LO level at RF input	
< 2.7 GHz	< – 90 dBm
> 2.7 GHz	< – 80 dBm

Software Features and Benchmarks

Description	Nominal Specification
E3238S Software Features	See the E3238S Configuration and Performance guide, http://www.agilent.com/find/n6820e.
Search Speed	
Sweep rate (1 Average, two E9821A Option 101 dual processor modules)	
136 Hz RBW	300 MHz/second
1.1 kHz RBW	640 MHz/second
17 kHz RBW	750 MHz/second
139 kHz RBW	760 MHz/second

General Specifications

Signal Surveyor Standard Configurations

VXI Mainframe Option	Total VXI Power Consumption	VXI System Weight	VXI Supply Voltage Range	VXI Supply Frequency Range	VXI Overall Dimensions HxWxD mm (in)
5 Slot MFRAME1	340W	19kg(41 lbs)	100-240 Vac	50-60 Hz	117 x 405 x 595 (7.0x16.0x23.4)
6 Slot E1421B	420W	24 kg (52 lbs)	90-264 Vac 90-132 Vac	45-66 Hz 360-440 Hz	234x450x600 (9.2x17.7x23.6)
13 Slot E8403/4A	380W	34 kg (74 lbs)	90-264 Vac 90-132 Vac	47-66 Hz 360-440 Hz	370x425x670 (14.6x16.7x26.4)

Description	Nominal Specification
Environmental	
Operating temperature range	0 to 40 °C[1]
Storage temperature range	– 40 to 70 °C[1]
Humidity	10 to 90% at 40 °C
Maximum Altitude	3,000 m (2,000 m using Agilent E1421B VXI Mainframe)
Regulatory Compliance	
Safety	The E3238S/N6820E Complies with European Low Voltage Directive 73/23/EEC, amended by 93/68/EEC: • IEC 61010-1:2001 / EN 61010-1:2001
EMC[2]	Selected E3238S/N6820E Systems are tested for Compliance with European EMC Directive 2004/108/EC. IEC 61326-2-1:2005 / EN 61326-2-1:2006 IEC 61326-1:2005 / EN 61326-1:2006 CISPR Pub 11 Group 1, class A ICES/NMB-001 This ISM device complies with Canadian ICES-001. Cet appareil ISM est conforme à la norme NMB-001 du Canada. The E3238S/N6820E passes IEC/EN 61326-1 when the front cover and ground wire are installed per instructions. Consult the Declaration of Conformity for the "E3238S/N6820E" to determine the compliance status and performance of specific hardware combinations. Contact your local Agilent Technologies office or sales representative for more information. See also http://regulations.corporate.agilent.com/

[1]Temperature range does not include any notebook computer restrictions, and may vary depending on the VXI mainframe model used. See http://www.agilent.com/find/n6820e.

[2]Must install the EMI and Cable Protection Kit to be compliant. See the EMI and Cable Protection Kit Installation Note that was included in your shipment.

Safety Summary

The following general safety precautions must be observed during all phases of operation, service, and repair of this product. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the product. Agilent Technologies, Inc. assumes no liability for the customer's failure to comply with these requirements.

Refer to the VXI Mainframe product documentation for specific instructions.

WARNINGS

Ground the equipment: For Safety Class 1 equipment (equipment having a protective earth terminal), an uninterruptible safety earth ground must be provided from the mains power source to the product input wiring terminals or supplied power cable.

DO NOT operate in an explosive atmosphere
Do not operate the instrument in the presence of flammable gases or flames.

For continued protection against fire, replace the line fuse(s) only with fuse(s) of the same voltage and current rating and type. DO NOT use repaired fuses or short-circuited fuse holders. Refer to the operation and service documentation for the VXI mainframe for information on the power supply specifications, grounding, and ventilation requirements.

Keep away from live circuits: Operating personnel must not remove equipment covers or shields. Procedures involving the removal of covers or shields are for use by service-trained personnel only. Under certain conditions, dangerous voltages may exist even with the equipment switched off. To avoid dangerous electrical shock, DO NOT perform procedures involving cover or shield removal unless you are qualified to do so.

DO NOT operate damaged equipment: Whenever it is possible that the safety protection features built into this product have been impaired, either through physical damage, excessive moisture, or any other reason, REMOVE POWER and do not use the product until safe operation can be verified by service-trained personnel. If necessary, return the product to an Agilent Technologies Sales and Service Office for service and repair to ensure that safety features are maintained.

If the equipment is used in a manner not specified by Agilent Technologies, the protection provided by the equipment may be impaired.

DO NOT service or adjust alone: Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.

DO NOT substitute parts or modify equipment: Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification to the product. Return the product to a Agilent Sales and Service Office for service and repair to ensure that safety features are maintained.

Cleaning: To prevent electrical shock, disconnect the system from the mains power before cleaning. Us a dry cloth or one slightly dampened with water to clean the external case parts. Do not attempt to clean internally.

Caution

DO NOT remove the module covers.
Operating personnel must not remove module covers. Component replacement and internal adjustments must be made only by qualified service personnel. Modules that appear damaged or defective should be made inoperative and secured against unintended operation until they can be repaired by qualified service personnel.

Laser Safety

This product contains Finisar 850nm SFF transceivers that are Class I laser products and comply with Laser Safety requirements per FDA/CDRH and IEC-825-1 regulations. Agilent Technologies, Inc. uses these products in compliance with FDA Laser Notice 42. The Finisar transceivers are certified per the following standards:

Finisar Transceiver Safety Specifications

Feature	Agency	Standard	Certificate Number
Laser Eye Safety	FDA/CDRH	FDA 21(J) CFR	9210176-17
Laser Eye Safety	TÜV	EN60950 EN 60825-1 EN 60825-2	R9772230.07
Electrical Safety	UL/CSA	CLASS 3862.07 CLASS 3862.87	CSA 1034405

Safety Symbols and Instrument Markings

Symbols and markings in manuals and on instruments alert you to potential risks, provide information about conditions, and comply with international regulations. The following table defines the symbols and markings you may find in a manual or on an instrument.

Safety symbols and instrument markings

Safety symbols	
	Warning: risk of electric shock.
	Caution: refer to accompanying documents.
	Alternating current.
	Earth (ground) terminal
	Protective earth (ground) terminal
	Frame or chassis terminal
	Terminal is at earth potential. Used for measurement and control circuits designed to be operated with one terminal at earth potential.
Instrument markings	
	The CE mark is a registered trademark of the European Community. If it is accompanied by a year, it indicates the year the design was proven.
 N10149	The C-tick mark is a registered trademark of the Spectrum Management Agency of Australia. This signifies compliance with the Australian EMC Framework regulations under the terms of the Radio Communications Act of 1992.
ISM1-A	This text indicates that the instrument is an Industrial Scientific and Medical Group 1 Class A product.
	This product complies with the WEEE Directive (2002/96/EC) marking requirements. The affixed label indicates that you must not discard this electrical/electronic product in domestic household waste.[1]

[1]The URL for take-back/WEEE information is http://www.agilent.com/environment/product

Service and Support

Any adjustment, maintenance, or repair of this product must be performed by qualified personnel. Contact your customer engineer through your local Agilent Technologies Service Center.

Agilent on the Web

You can find information about technical and professional services, product support, and equipment repair and service on the Web at:

http://www.agilent.com/

1. Click the Select a Country or Area link (upper-right corner of the page) and select your country. When you click "Submit" it returns to the Agilent home.
2. Hover over the Products and Services tab.
3. Select the Repair and Calibration Services link.

Agilent by Phone

Or you can call one of the numbers in the following table.

Agilent Call Centers and Regional Headquarters

United States and Canada:	Test and Measurement Call Center (800) 452 4844 (toll-free in US)
Europe:	(41 22) 780 8111
Japan:	Measurement Assistance Center (81) 0426 56 7832
Latin America:	305 269 7548
Asia-Pacific:	(85 22) 599 7777

Index

Appendix A: d.E3238s Listing

The following file listing is from the d.E3238s Motif resource file in the E3238s home directory. *Do not modify this file.* To modify or add resource parameters, copy it to the name E3238s and modify that file.

```
!! $Revision: 1.14 $
!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! Application Resources    !!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*openScreenTime:           5
*useHardware:              True
*newEnergyLogViewLength:   1000
*handoffLogViewLength:     500
*alarmLogViewLength:       100
*spectrogramBackingStoreSize: 2000000
*dataBufferSize:           8000000
*maxEnergyHistorySize:     5000
*signalDatabaseSize:       500000
*sr71Enabled:              False
*enhancedSpectrogramMarker:    True
*maxSpectrogramColors:     32
*displayLocalTime:         True
*maxHandoffRxs:            16
```

Appendix B: d.e3238s.cfg Listing

When the E3238S program starts, the system looks for a file named `e3238s.cfg`. If the location of the file is not specifically defined in the resource file (see Application Resources (pg 183), the configuration file in the `e3238s` directory is used.

The file installed initially is `d.e3238s.cfg` in the `e3238s` directory. If no file exists named `e3238s.cfg`, a duplicate of `d.e3238s.cfg` is created and given that name.

When you upgrade (i.e., install the E3238S software on a system with an earlier version) the installation process does *not* overwrite the existing `e3238s.cfg` file. This avoids destroying configuration information you may need. You may need to edit the file and incorporate new commands from the `d.e3238s.cfg` file before the upgrade will perform properly.

The commands in this file are documented in the chapter beginning on page 183. The following is a listing of the `d.e3238s.cfg` file.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!  E3238s Hardware Configuration Resource File    $Revision: 1.59 $    !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! This file is initially installed in the E3238s Signals Development
! System as d.e3238s.cfg in the C:\E3238s\ directory.  It is
! automatically copied to C:\E3238s\e3238s.cfg at installation time
! if the file e3238s.cfg does not already exist.
!
! This file is used to tell the software what type of hardware to
! configure into the system.  The file configures the switches,
! tuners, ADC's modules, DSP modules, and handoff receivers.  The menu
! access setup is also configured in this file.
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                         SICL Configuration                              !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! The system needs to know what vxi interface to use.
!
vxiInterface:                   VXI0
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                  Search Receiver Switch Configuration                     !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! You can select one antenna from a list of inputs with an RF switch.
! This software supports the E1472A (RF) and E1368/69/70 (microwave) VXI
! switch modules.  Only one module may be used at a time and, if
! multiple tuners are used, the switches of the one module must be
! used both to select antennas and tuners.  See the Configure section in
! the Operator's Reference for details.
!
! Usage of the E1472A module is described first, then the usage of the
! E1368/69/70.  If no antenna switching is being used, set the
! 'switchModel' parameter to 'None'.  Note that the E1474A module is code
! compatible with the E1472, but has 75 ohm inputs.
!
! For 'switchModel: E1472A'
!  - 'switchInterfaceParm' specifies the logical address of the switch
!      module.
!  - 'switchConfiguration' specifies up to a 31 character string that
!     appears in the switch configuration dialog box to describe the type
!     of switch configuration.
!  - 'switchConnections' specifies the labels for each of the switch's
!     input and output connections.  Each input/output connection label
!     is separated by a comma.  The order of each field for the E1472A
!     switch is as follows:
!
!             00, 01, Com 00, 02, 03, (Bank 0)
!             10, 11, Com 10, 12, 13, (Bank 1)
!             20, 21, Com 20, 22, 23, (Bank 2)
!             30, 31, Com 30, 32, 33, (Bank 3)
!             40, 41, Com 40, 42, 43, (Bank 4)
!             50, 51, Com 50, 52, 53  (Bank 5)
!
! Each of these labels can be up to 16 characters and the entire length
! of the switchConnections string has to be less than 256 characters.
```

```
! When the label is in the form @A1, @A2 ... @A16, this string is
! replaced by the corresponding antenna name.
!
! The 'switchCmnd' and 'switchMask' commands specify which switch relays
! must be updated to connect the particular antenna input to the tuner
! such that switch activity is minimized to maximize contact life.
!
! The values are hexadecimal numbers where each bit specifies which of
! the selectable connections is connected to the common connection.
! The number of hex "digits" corresponds to the number of switch banks.
! The bits are mapped as follows:
!
! Input/Output                                    Input/Output
!     30--------------------------+ +----------------23
!     31--------------------------+| |+---------------22
!     32--------------------------+|| ||+--------------21
!     33--------------------------+|| |||+-------------20
!                                  ||| ||||
!     40----------------------+    ||| ||||  +-----------13
!     41----------------------+|   ||| |||| |+-----------12
!     42----------------------+||  ||| |||| ||+----------11
!     43----------------------+||| ||| |||| |||+---------10
!                             |||| |||| |||| ||||
!     51-----------------+    |||| |||| |||| ||||  +------03
!     52-----------------+|   |||| |||| |||| |||| |+-----02
!     53-----------------+||  |||| |||| |||| |||| ||+----01
!     54-----------------+||| |||| |||| |||| |||| |||+---00
!                        |||| |||| |||| |||| |||| ||||
!                        vvvv vvvv vvvv vvvv vvvv vvvv
!          0000 0000 XXXX XXXX XXXX XXXX XXXX XXXX
!
! See the discussion of the switchCmnd command in the Configuration
! section of the Command Reference for more information.
!
!searchRx1.switchModel:         E1472A
!searchRx1.switchInterfaceParm: 131
!searchRx1.switchConfiguration: 16 : 1
!searchRx1.switchConnections: \
!  @A1,@A2,To 40,@A3,@A4,\
!  @A5,@A6,To 41,@A7,@A8,\
!  @A9,@A10,To 42,@A11,@A12,\
!  @A13,@A14,To 43,@A15,@A16,\
!  From Com 00,From Com 10,To Tuner,\
!  From Com 20,From Com 30,\
!  ,,,,,,,,
!searchRx1.antenna1.switchCmnd:         0x00010001
!searchRx1.antenna1.switchMask:         0x000F000F
!searchRx1.antenna2.switchCmnd:         0x00010002
!searchRx1.antenna2.switchMask:         0x000F000F
!searchRx1.antenna3.switchCmnd:         0x00010004
!searchRx1.antenna3.switchMask:         0x000F000F
!searchRx1.antenna4.switchCmnd:         0x00010008
!searchRx1.antenna4.switchMask:         0x000F000F
!searchRx1.antenna5.switchCmnd:         0x00020010
!searchRx1.antenna5.switchMask:         0x000F00F0
!searchRx1.antenna6.switchCmnd:         0x00020020
!searchRx1.antenna6.switchMask:         0x000F00F0
!searchRx1.antenna7.switchCmnd:         0x00020040
!searchRx1.antenna7.switchMask:         0x000F00F0
!searchRx1.antenna8.switchCmnd:         0x00020080
!searchRx1.antenna8.switchMask:         0x000F00F0
!searchRx1.antenna9.switchCmnd:         0x00040100
!searchRx1.antenna9.switchMask:         0x000F0F00
!searchRx1.antenna10.switchCmnd:        0x00040200
!searchRx1.antenna10.switchMask:        0x000F0F00
!searchRx1.antenna11.switchCmnd:        0x00040400
!searchRx1.antenna11.switchMask:        0x000F0F00
!searchRx1.antenna12.switchCmnd:        0x00040800
!searchRx1.antenna12.switchMask:        0x000F0F00
!searchRx1.antenna13.switchCmnd:        0x00081000
!searchRx1.antenna13.switchMask:        0x000FF000
!searchRx1.antenna14.switchCmnd:        0x00082000
!searchRx1.antenna14.switchMask:        0x000FF000
!searchRx1.antenna15.switchCmnd:        0x00084000
!searchRx1.antenna15.switchMask:        0x000FF000
!searchRx1.antenna16.switchCmnd:        0x00088000
!searchRx1.antenna16.switchMask:        0x000FF000
!
!
! For 'switchModel: E1368/69/70A'
!  - 'switchInterfaceParm' specifies the logical address of the switch
!      module.
!  - 'switchConfiguration' specifies up to a 31 character string that
!     appears in the switch configuration dialog box to describe the type
```

```
!     of switch configuration.
!  - 'switchConnections' specifies the labels for each of the switch's
!     input and output connections.  Each input/output connection label
!     is separated by a comma.  The order of each field for the
!     E1368/69/70A switch is as follows:
!
!             01, Com 00, 02, (Bank 0)
!             11, Com 10, 12, (Bank 1)
!             21, Com 20, 22, (Bank 2)
!
! Each of these labels can be up to 16 characters and the entire length
! of the switchConnections string has to be less than 256 characters.
! If the label is in the form @A1, @A2 ... @A16, this string is
! replaced by the corresponding antenna name.
!
! The 'switchCmnd' and 'switchMask' commands specify which switch relays
! must be updated to connect the particular antenna input to the tuner
! such that switch activity is minimized to maximize contact life.
!
! The values are hexadecimal numbers where each bit specifies which of
! the selectable connections is connected to the common connection.
! The number of hex "digits" corresponds to the number of switch banks.
! See the discussion of the switchCmnd command in the Configuration
! section of the Command Reference.  The configuration of the switch
! module is discussed in the Hardware Installation Note.
!
!searchRx1.switchModel:         E1368/69/70A
!searchRx1.switchInterfaceParm: 131
!searchRx1.switchConfiguration: 4 : 1
!searchRx1.switchConnections:   @A1,To 21,@A2,\
!                               @A3,To 22,@A4,\
!                               From Com 00,To Tuner,From Com 10
!searchRx1.antenna1.switchCmnd:         0x00
!searchRx1.antenna1.switchMask:         0x05
!searchRx1.antenna2.switchCmnd:         0x01
!searchRx1.antenna2.switchMask:         0x05
!searchRx1.antenna3.switchCmnd:         0x04
!searchRx1.antenna3.switchMask:         0x06
!searchRx1.antenna4.switchCmnd:         0x06
!searchRx1.antenna4.switchMask:         0x06
!

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                   Multiple Tuner Switching                                  !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! The switch can also be used to switch the IF output from multiple
! tuners into the ADC module.  The 'tunerSwitchCmnd' and 'tunerSwitchMask'
! work exactly like the corresponding commands 'switchCmnd' and
! 'switchMask' for antenna switching.
!
! See the discussion of the tunerSwitchCmnd command in the Configuration
! section of the Command Reference.  The configuration of the switch
! module is discussed in the Hardware Installation Note.  See also,
! the Multiple Tuner Configuration discussion later in this file.
!
!searchRx1.tuner1.tunerSwitchCmnd:      0x00100000
!searchRx1.tuner1.tunerSwitchMask:      0x00F00000
!searchRx1.tuner2.tunerSwitchCmnd:      0x00200000
!searchRx1.tuner2.tunerSwitchMask:      0x00F00000
!searchRx1.tuner3.tunerSwitchCmnd:      0x00400000
!searchRx1.tuner3.tunerSwitchMask:      0x00F00000
!searchRx1.tuner4.tunerSwitchCmnd:      0x00800000
!searchRx1.tuner4.tunerSwitchMask:      0x00F00000
!

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                Search Receiver Tuner Configuration                          !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! This software supports the following 'tunerModels':
!   HP89430A:       2.0 -  1800 MHz
!   HP89431A:       2.0 -  2650 MHz
!   WJ9119:         0.5 -    32 MHz
!   WJ9119-1:       1.0 -    32 MHz
!   E2730:         20.0 -  2700 MHz  With E1439 or N6830A/70 ADC
!   E2731:         20.0 -  6000 MHz  With E1439 or N6830A/70 ADC
!   CS5020:       500.0 - 20000 MHz  Only with E1439 ADC [60 GHz with options]
!   CS5040:       500.0 - 20000 MHz  Only with E1439 ADC [60 GHz with options]
!   SI9136:        20.0 -  3000 MHz  With E1439 or N6830A/70 ADC
!   ADV3000:       20.0 -  3000 MHz  Only with E1439 ADC
!   PSA:            0.1 - 26500 MHz  E4440A with HY7. With E1439 or N6830A/70 ADC
!   N6830A/HF       0.1 -    32 MHz  Only with N6830A/HF ADC
!
! The following combinations of block downconverters and tuners are
! supported as these 'tunerModels' :
```

```
!   SI9250-E273X:    20 - 18000 MHz  With E1439 ADC and E2730/31 or,
!                                      N6830A/70 ADC and E2730/31
!   SI9250-ADV3000:  20 - 18000 MHz  Only with E1439 ADC and ADV3000
!   SI9250-SI9136:   20 - 18000 MHz  With E1439 ADC and DRS 9136B or,
!                                      N6830A/70 ADC and DRS 9136B
!   CS5320:        2000 - 18000 MHz  Only with E1439 ADC [to 60 GHz]
!
! When no tuner is used, the tuner model is None and the output of the
! switch or antenna should be connected directly to the ADC module.
!
!
!!!!!!!!!!!!!!!!!!!!!!!!
! HP89430A / HP89431A
!!!!!!!!!!!!!!!!!!!!!!!!
! The 'tunerInterfaceParm' for the  HP89430A or HP89431A is the RS-232
! baud rate. The baudrate should be 125000. Optional I & Q values can be
! specified after the baudrate (baudrate, I, Q) to minimize the LO feed-thru
! for the 89431A. The I & Q values for your 89431A can be found at:
! URL:  ftp://ftp.agilent.com/pub/dsp/products/89431/LO-I-Q-Values.xls
! Email: eveswl@agilent.com.  Phone:  (425) 356-6261   FAX:  (425) 356-6260
! You will need to provide the serial # of your 89431A.
!
! 89430 parameters: baud rate
!searchRx1.tuner1.tunerModel:          HP89430A
!searchRx1.tuner1.tunerInterfaceParm:  125000
!
!
! 89431 parameters: baud rate
!searchRx1.tuner1.tunerModel:          HP89431A
!searchRx1.tuner1.tunerInterfaceParm:  125000
!
!!!!!!!!!!!!!!!!!!!!!!!!
! WJ9119 / WJ9119-1
!!!!!!!!!!!!!!!!!!!!!!!!
! The 'tunerInterfaceParm' for the WJ9119 is a comma-separated value string
! that specifies:
!   LO module logical address    1 to 255
!   RF module logical address    1 to 255
!   10 MHz Reference             0 (Internal) or 1 (External)
!   ADC Clock Output             1 to 255
!   1st LO Output                1 to 255
!   2nd LO Output                1 to 255
! The last three parameters are used to set which of the 8 outputs are to
! be enabled on the front panel of the LO.  The valued entered is a mask
! for which outputs are enabled.  An example follows:
!    1: Enable output 1.
!    7: Enable output 1, 2, and 3.
!  255: Enable all outputs.
! An additional two optional parameters can be passed to specify the settling
! time for a small LO step (< 7.5 MHz) and a large LO step (> 7.5 MHz).  The
! settling times are specified in microseconds with both values having a
! default of 500 microseconds.
!
! WJ9119 parameters: LO logical address, RF logical address, 10 MHZ Ref,
!                    ADC clock output, 1st LO output, 2nd LO output
!                    optional small settle time, large settle time in us
!searchRx1.tuner1.tunerModel:          WJ9119
!searchRx1.tuner1.tunerInterfaceParm:  140, 142, 0, 0, 1, 1
!searchRx1.tuner1.tunerInterfaceParm:  140, 142, 0, 0, 1, 1, 500, 500
!
!
! WJ9119-1 parameters: LO logical address, RF logical address, 10 MHZ Ref,
!                      ADC clock output, 1st LO output, 2nd LO output
!                      optional small settle time, large settle time in us
!searchRx1.tuner1.tunerModel:          WJ9119-1
!searchRx1.tuner1.tunerInterfaceParm:  140, 142, 0, 0, 1, 1
!searchRx1.tuner1.tunerInterfaceParm:  140, 142, 0, 0, 1, 1, 500, 500
!
!
!!!!!!!!!!!!!!!!!!!!!!!!
! Agilent E2730A / E2730B
! Agilent E2731A / E2731B
!!!!!!!!!!!!!!!!!!!!!!!!
! The 'tunerInterfaceParm' for the E2730A / E2730B / E2731A / E2731B tuner is
! a comma-separated value string that specifies:
!   Logical Address     1 to 255
!   10 MHz Reference    0 (Internal), 1 (External), or 2 (VXI Backplane)
!   Settling Time       1 to 1000 mSec   Typical Range: 3 to 5 mSec
!   Start Frequency     2 MHz to 20 MHz  Default: 20 MHz
!   89605 Logical Addr  Optional, 1 to 255
!
! The adcModel can be E1439x/70 or N6830A/70
!
searchRx1.tuner1.tunerModel:          E2730A
```

```
searchRx1.tuner1.tunerInterfaceParm:  136, 0, 3, 20
!
!searchRx1.tuner1.tunerModel:          E2731A
!searchRx1.tuner1.tunerInterfaceParm:  136, 0, 3, 20
!
!!!!!!!!!!!!!!!!!!!!!!!!
! CS5040 or CS5020
!!!!!!!!!!!!!!!!!!!!!!!!
! The 'tunerInterfaceParm' for the CS5040 tuner is a comma-separated value string
! that specifies:
!   Logical address     1 to 255   (or baud rate for CS5020C)
!   Settling Time       1 to 1000 mSec   Typical Range: 5 to 25 mSec
!   Start Frequency     500 MHz
!   Stop Frequency      20000 MHz (no downconverter)
!                       40000 MHz (CS-5040-K/Ka)
!                       60000 MHz (CS-5040-K/Ka and CS-5040-U)
!
!searchRx1.tuner1.tunerModel:          CS5040
!searchRx1.tuner1.tunerInterfaceParm:  142,   10, 500, 20000
!
!searchRx1.tuner1.tunerModel:          CS5020
!searchRx1.tuner1.tunerInterfaceParm:  38400, 25, 500, 20000
!
!!!!!!!!!!!!!!!!!!!!!!!!
! CS5320 & E273X
!!!!!!!!!!!!!!!!!!!!!!!!
! The 'tunerInterfaceParm' for the CS5320 tuner is a comma-separated value string
! that specifies:
!   CS5320 Baudrate         19200 or 38400
!   CS5320 Bandwidth        250 to 500 MHz
!   CS5320 Stop Frequency   18000 MHz (no downconverter)
!                           40000 MHz (CS-5320A-K/Ka)
!                           60000 MHz (CS-5320A-K/Ka and CS-5320A-U)
!   CS5320 Short Settle     1 to 1000 mSec   Typical: 25 mSec
!   CS5320 Long Settle      1 to 1000 mSec   Typical: 60 mSec
!   E273X Logical address   1 to 255
!   E273X 10 MHz Reference  0 (Internal) or 1 (External)
!   E273X Settling Time     1 to 1000 mSec   Typical Range: 3 to 5 mSec
!
!searchRx1.tuner1.tunerModel:   CS5320A
!searchRx1.tuner1.tunerInterfaceParm:  19200, 500, 18000, 25, 60, 136, 1, 3
!
!
!!!!!!!!!!!!!!!!!!!!!!!!
! SI9250 & E273X
!!!!!!!!!!!!!!!!!!!!!!!!
! DRS (Signia-IDT) 9250 18GHz block downconverter and E2730 or E2731 tuner.
! The adcModel must be E1439x/70 or N6830A/70.
! The SI9250 requires an external frequency reference.
! The 'tunerInterfaceParm' for the this converter/tuner specifies:
!   SI9250 Logical Address  1 to 255
!   SI9250 Reference        0 (Ext 10MHz) or 1 (Ext 100MHz)
!   SI9250 Settling Time    1 to 1000 mSec Typical: 10 mSec
!   E273X Logical Address   1 to 255
!   E273X 10 MHz Reference  0 (Internal) or 1 (External)
!   E273X Settling Time     1 to 1000 mSec   Typical Range: 3 to 5 mSec
!   E273X RF Atten dB       0 to 20 dB  Typical Range: 0 dB
!
!searchRx1.tuner1.tunerModel:   SI9250-E273X
!searchRx1.tuner1.tunerInterfaceParm: 144, 0, 10, 136, 0, 3, 0
!
!!!!!!!!!!!!!!!!!!!!!!!!
! DRS SI 9136
!!!!!!!!!!!!!!!!!!!!!!!!
! DRS (Signia-IDT) 9136B dual VME tuner. Caution: Signals on the VME
! P2 connector of the 9136B are not compatible with the VXI
! backplane.  To avoid damage, ribbon cable (J2) must be removed
! from the E1407 carrier module.
! The 'tunerInterfaceParm' for the SI9136 tuner:
!   VME Address             0 to 48896 in steps of 2048. Typical: 4096
!   Channel                 1 or 2
!   LO Mode                 0 (Independent), 1 (Master), 2 (Slave)
!   10 MHz Reference        0 (Internal), 1 (External 10 MHz)
!   Start Frequency         2 MHz to 20 MHz  Default: 5 MHz
!   Preselector Filter      0 (bypass), 1 (enable)
!   Settling                100 to 10000 uSec (400 typical)
!
! Additional info:
!   VME address:  This is the A16 VME bus address, not a VXI logical
!       address. Enter the decimal equivalent. Hex 0x1000 is 4096.
!       To avoid conflicts with VXI, do not use addresses above 0xC000.
!       To set the module to A16 address 0x1000, set all S1 switches
!       0 (handle toward board edge) except switch S1-4.
!   Channel:  Each module provides 2 search tuners. To use Ch2,
```

```
!       add second tunerInterfaceParm line with the same VME address
!       and 2 for the channel.
!   LO Mode:  Controls sharing of the 1st and 2nd LO. (0=independent,
!       1=Master, 2=Slave).
!   Preselector Filter:  Enables tuned preselection filter when tuned
!       between 20 MHz and 125 MHz.  Limits IF bandwidth to a few MHz.
!
!searchRx1.tuner1.tunerModel:          SI9136
!searchRx1.tuner1.tunerInterfaceParm:  4096, 1,  0,  0, 5, 0, 400
!searchRx2.tuner1.tunerModel:          SI9136
!searchRx2.tuner1.tunerInterfaceParm:  4096, 2,  0,  0, 5, 0, 400
!
!!!!!!!!!!!!!!!!!!!!!!!
! DRS SI 9250 and SI 9136
!!!!!!!!!!!!!!!!!!!!!!!
! DRS (Signia-IDT) 9250-2 18GHz block downconverter and a DRS
! 9136B dual VME tuner. The adcModel must be E1439x/70 or N6830A/70.
! The SI9250 requires an external frequency reference.
! See instructions above for setting 9250 and 9136B VME address.
! The 'tunerInterfaceParm' string that specifies:
!   SI9250 Logical Address  1 to 255
!   SI9250 Reference        0 (Ext 10MHz) or 1 (Ext 100MHz)
!   SI9250 Settling Time    1 to 1000 mSec Typical: 10 mSec
!   9136 VME Address        0 to 48896 in steps of 2048. Typ: 4096
!   9136 Channel            1 or 2
!   9136 LO Mode            0 (Independent)
!   9136 10 MHz Reference   0 (Internal), 1 (External 10 MHz)
!   Start Frequency         2 MHz to 20 MHz  Default: 20 MHz
!   Settling                100 to 10000 uSec (400 typical)
!
!searchRx1.tuner1.tunerModel:          SI9250-SI9136
!searchRx1.tuner1.tunerInterfaceParm:  144, 0, 10, 4096, 1,  0,  0, 20, 400
!searchRx2.tuner1.tunerModel:          SI9250-SI9136
!searchRx2.tuner1.tunerInterfaceParm:  144, 0, 10, 4096, 2,  0,  0, 20, 400
!
!!!!!!!!!!!!!!!!!!!!!!!
! Mercury Echotek ADV-3000T
!!!!!!!!!!!!!!!!!!!!!!!
! The 'tunerInterfaceParm' for the ADV-3000T tuner is a comma-separated
! value string that specifies:
!   VME Address             0 to 48896 in steps of 256. Typical: 256
!   10 MHz Reference        0 (Internal), 1 (External 10 MHz)
!   Switch Frequency (MHz)  0 to 3000. Typical: 0
!   IF Atten dB             0 to 35 dB in 5 dB steps. Typical: 35
!
! Additional info:
!   VME address:  This is the VME bus address, not a VXI logical address,
!       of the Synthesizer LO module of this two-module tuner.
!       Enter the decimal equivalent. Hex 0x0100 is decimal 256.
!       The module address is set with switches on the module.
!       The switch nearest the VME connector sets the 'Y' digit in
!       the hex address 0xYZ00. The adjacent switch sets the 'Z' digit.
!       To avoid conflicts with VXI, do not use addresses above 0xC000.
!       The RF tuner module address must be set one switch setting
!       higher (256 VME addresses higher) than the LO module.
!   Switch Frequency:  entering a non-zero  value will cause RF2 input
!       to be automatically selected for frequencies above the value.
!
! searchRx1.tuner1.tunerModel:          ADV3000
! searchRx1.tuner1.tunerInterfaceParm:  256, 0,    0,  35
!
!!!!!!!!!!!!!!!!!!!!!!!
! SI9250 & ADV3000
!!!!!!!!!!!!!!!!!!!!!!!
! DRS (Signia-IDT) 9250-2 18GHz block downconverter and a Mercury
! ADV3000T tuner. The adcModel must be E1439x/70
! The SI9250 requires an external frequency reference.
! See instructions above for setting ADV3000 VME address.
! The 'tunerInterfaceParm' for the SI9250-ADV3000 converter/tuner specifies:
!   SI9250 Logical Address  1 to 255
!   SI9250 Reference        0 (Ext 10MHz) or 1 (Ext 100MHz)
!   SI9250 Settling Time    1 to 1000 mSec Typical: 10 mSec
!   ADV3000 VME Address     0 to 48896 in steps of 256. Typical: 256
!   ADV3000 Reference       0 (Internal), 1 (External 10 MHz)
!   ADV3000 RF Atten dB     -14 to 36 dB in 2 dB steps. Typical: 10
!
! searchRx1.tuner1.tunerModel:   SI9250-ADV3000
! searchRx1.tuner1.tunerInterfaceParm:  144, 0, 10, 256, 1, 10
!
!
!!!!!!!!!!!!!!!!!!!!!!!
! PSA Series Spectrum Analyzer
!!!!!!!!!!!!!!!!!!!!!!!
! Agilent E4440A, E4443A, E4445A, E4446A, E4448A Spectrum Analyzer.
```

```
! PSA must have option HY7 or option H70, which adds 70 MHz IF output.
! The IF output should be connected to E1439 ADC Analog In or N6830A 70MHz
! IF input.  The E9821 controls the PSA via the LAN.  The LAN interface must
! be configured and connected on both the PSA and the E9821.  To
! configure the E9821 LAN interface, run e9821LanConfigure.exe.
!
! Enabling the PSA option preamplifier improves sensitivity
! but will limit the frequency range. Option 1DS preamp
! is 100kHz to 3 GHz, option 110 preamp is 10 MHz to 50 GHz.
!
! The 'tunerInterfaceParm' for the PSA 'tuner' specifies:
!   PSA LAN IP address      dotted decimal IP address or hostname
!   PSA Stop Frequency      3000 to 50000 MHz
!   PSA Reference           0 (Internal) or 1 (External)
!   PSA Settling Time       1 to 1000 mSec Typical: 45 mSec
!   PSA Preamplifier        0 (off) or 1 (Opt 1DS on) or 110 (Opt 110 on)
!   PSA IF gain (milli-dB)  9000 (+9dB for HY7) or -6000 (for H70)
!
! searchRx1.tuner1.tunerModel:   PSA
! searchRx1.tuner1.tunerInterfaceParm:  192.168.0.3, 26500, 0, 45, 0, 9000
!
!
!!!!!!!!!!!!!!!!!!!!!!!
! None - No Tuner
!!!!!!!!!!!!!!!!!!!!!!!
! No Tuner
!searchRx1.tuner1.tunerModel:     None
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                        Multiple Tuner Configuration                        !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Switch modules can be used to implement as many as 4 tuners such that
! any one can be selected while the program runs (no restart required).
! The switch-defining commands are the same as used to select antennas.
! See the 'tunerSwitchCmnd' in the Configuration section of the Command
! Reference.
!
! Only one tuner of a specific type can be loaded simultaneously and the
! tuner specified in the 'tuner1' slot is the tuner the software wakes up
! using.  The tuner model 'Off' is used if the particular tuner slot is
! not active.
!
! The following is a multi-tuner configuration example:
!
!searchRx1.tuner1.tunerModel:          WJ9119-1
!searchRx1.tuner1.tunerInterfaceParm:  140, 142, 0, 1, 1, 1
!
!searchRx1.tuner2.tunerModel:          None
!searchRx1.tuner2.tunerInterfaceParm:
!
!searchRx1.tuner3.tunerModel:          HP89431A
!searchRx1.tuner3.tunerInterfaceParm:  125000
!
!searchRx1.tuner4.tunerModel:          Off
!searchRx1.tuner4.tunerInterfaceParm:
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                  Search Receiver ADC Configuration                         !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! The ADC configuration commands are as follows:
!
!  - adcModel [E1437A, E1438A, E1438B, E1439A/70, E1439A/BB, E1439B/70,
!              E1439B/BB,E1439C/70, E1439C/BB, E1439D/70, E1439D/BB]
!              N6830A/70, N6830A/HF]
!       Specifies the ADC's model type.
!
!       If the ADC model type is N6830A/HF, only the HF input of the N6830
!       module is used and the tunerType can be set to None.
!
!  - adcInterfaceParm  [1 to 254]
!       specifies the ADC's logical address.
!
!  - adcClock  [Internal, External]
!       specifies whether to use an internal or external ADC clock.
!       The clock must be 20.48 MHz for the E1437A module.  For the E1438,
!       E1439, or N6830 modules, the frequency must be 10 MHz, and it will be used
!       as a reference to derive the actual sample clock.  The clock must
!       be present at the clock input before the software is started.
!
!  - adcDataPort  [localBus, FPDP, VXI]
!       Specifies the data path between the ADC and DSP module. The N6830A
!       module only supports FPDP
```

```
!
!  - adcMasterClock [Off, On, Auto]
!       Specifies whether the ADC puts its clock signal on the VXI backplane
!       for use by other modules. Only one ADC may drive the VXI backplane. If
!       the Auto mode is selected, the adcMasterClock for the search ADC is
!       turned Off unless the searchRxConfiguration is set to a multiple
!       channel mode.  In this case the first ADC is set to On and the other
!       ADC's are set to off.
!
!       When set to ON, the N6830A outputs a reference clock on the Ref Out
!       front panel SMB connector.  It does not output a clock on the VXI
!       backplane.
!
!  - adcSampleRate [10240000, 20480000, 40960000, 81920000, etc (see below)]
!       Specifies the ADC sample rate for the N6830A.  This parameter will
!       affect the bandwidth available for narrowband signal processing.
!
!       For best probability of intercept (fastest search revisit times) use
!       the lowest stare bandwidth that covers the frequency range of
!       interest.
!
!         N6830A/HF
!         Sample Rate      Stare Bandwidth
!         81920000         32 MHz
!         40960000         16 MHz
!         20480000          8 MHz
!         10240000          4 MHz
!
!         N6830A/70
!         Sample Rate      Stare Bandwidth
!         95000000         36 MHz
!         47500000         18 MHz
!         23750000          9 MHz
!         11870000        4.5 MHz
!
!       If you are upgrading from a 9119-1 tuner with E1437 ADC
!       to an N6830A/HF, use the following settings.
!
!         searchRx1.adcModel:      N6830A/HF
!         searchRx1.adcSampleRate: 20480000
!         searchRx1.adcDataPort:   FPDP
!
!       Increasing the ADC sample rate may affect the sweep rate depending
!       on the search setup number of averages and RBW selections.
!
searchRx1.adcModel:            E1439D/70
searchRx1.adcInterfaceParm:    130
searchRx1.adcClock:            Internal
searchRx1.adcDataPort:         FPDP
searchRx1.adcMasterClock:      Auto
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                  Search Receiver DSP Configuration                          !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! This software supports the E9821A DSP module.
!
! The DSP configuration commands are as follows:
!  - model  [E9821A]
!       specifies the DSP's model type.
!
!  - dspModules  [128]
!       specifies a comma separated list of DSP modules logical addresses.
!
!  - dspNoHardwareConfig
!       specifies the DSP hardware when the -noHardware option is used. The
!       choices are:
!          2: Dual G4 PMC
!          1: Single G4 PMC
!          D: Digital Downconverter PMC
!          E: Empty
!
!       examples:
!          222D       Three dual G4's and a DDC
!          DDD2       Three DDC's and a dual G4
!          222D DDD2  Two Modules with the first having three dual G4's
!                     and a DDC and the second having three DDC's and a
!                     dual G4
!
!  - dspCmndPort  [VXI]
!       specifies the command path between the DSP module and host computer.
!
!  - dspDataPort  [VXI]
!       specifies the data path between the DSP module and host computer.
```

```
!
!  - dspDataCompression [Off]
!       specifies whether to compress the frequency data uploaded from the
!       DSP module. The compression causes approximately +/- 0.1 dB
!       reduction in amplitude accuracy.
!
!  - downloadable  [C:\E3238s\downloadables\e9821a.out]
!       specifies the DSP program.
!
searchRx1.dspModel:            E9821A
searchRx1.dspModules:          128
searchRx1.dspNoHardwareConfig: 222D
searchRx1.dspCmndPort:         VXI
searchRx1.dspDataPort:         VXI
searchRx1.dspDataCompression:  On
downloadable:                  C:\E3238s\downloadables\e9821a.out
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!           Wideband Record/Playback Configuration                            !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!     wrpIpAddress:            LAN IP address of the Conduant Data
!                              recorder in dot notation.
!
!searchRx1.wrpIpAddress:       10.1.249.101
!
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                   IRIG Timing Configuration                                 !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! These commands define the system time reference configuration.
! For an overview of IRIG timing settings, see the Configure section of
! the Operator Reference.  See also, the Configuration section of the
! Command Reference.
!
!     timeReference:           [systemClock, vxiIRIG]
!     timeReferenceCal:        [0, 1] Enable adjustment of ADC sample
!                                     clock generated timestamps to track
!                                     IRIG time. Only applies with tuner
!                                     locked and vxiIRIG timeReference.
!     irigModel:               [None, BC350VXI]
!     irigInterfaceParm:       200
!     irigOperatingMode:       [Decode, Freerun, 1PPS, RealtimeClock,
!                                           GPSonBoard, GPSinAntenna]
!     irigTimeCodeFormat:      [IRIGA, IRIGB, 2137, NASA36, XR3]
!     irigTimeCodeModulation:  [AM, PCM]
!     irigClock:               [Internal, External]
!     irigSecondsFromGMT:      [-] Behind GMT  [+] Ahead GMT
!     irigCableDelay:          100 nanoseconds increments
!
!
timeReference:                 systemClock
timeReferenceCal:              1
irigModel:                     bc350vxi
irigInterfaceParm:             200
irigOperatingMode:             Freerun
irigTimeCodeFormat:            IRIGB
irigTimeCodeModulation:        AM
irigClock:                     Internal
irigSecondsFromGMT:            0
irigCableDelay:                0
!

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                  Handoff Receiver Configuration                             !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! The following handoff receivers are supported in this software:
!
!           None
!           WJ-8607
!           WJ-8611
!           WJ-8615P
!           WJ-8621
!           WJ-8629
!           WJ-8629A
!           WJ-8633-1
!           WJ-8634
!           WJ-8711
!           WJ-8712A
!           WJ-8712P
!           WJ-8721
!           WJ-8723
!           Cubic VXI-3250
!           Cubic VXI-3550
```

```
!           Cubic VXI-3570
!           Cubic R-2411/U
!           Cubic R-2412/U
!           ICOM R20
!           ICOM R8500
!           ICOM PCR1000
!           Ten-Tech RX-331
!           WiNRADiO G303 or G313
!           Software
!           Agilent RF Sensors (for example, the Agilent N6841A RF sensor)
!
! The software receiver receives handoffs and simulates a real
! handoff receiver.  No software detection or processing is done.
!
! The following analyzers are supported in this software:
!
!           Agilent 89441A
!           Agilent 89600
!
!
! Each handoff receiver is described with two lines in this file:
!
!   handoffRx<N>.driver:             C:\E3238s\lib\<driver shared library>
!   handoffRx<N>.interface:          <interface>,<interface parameters>
!
! where
!   <N> is the number of the handoff receiver that appears on the
!       e3238s user interface (e.g. 1), may be as large as 100
!   <driver shared library> is the filename of the handoff receiver
!       shared library (e.g. C:\E3238s\lib\HD_8711.dll),
!   <interface>,<interface parameters> identifies the hardware interface
!       to the receiver and parameters of the interface
!       (e.g rs232,ASRL0,9600).
!
! Examples for each supported handoff receiver are present further down
! in this file.
!
! The 'label' command is the label which appears in the handoff
! receiver pane.  It can be up to 31 characters in length.
!
!
!!!!!!!!!!!!!!!!!!!!!
!!! WJ-8607        !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRx1.driver:             C:\E3238s\lib\HD_8607.dll
!handoffRx1.interface:          rs232,ASRL0,9600
!handoffRx1.label:              VHF/UHF  Rx
!
!!!!!!!!!!!!!!!!!!!!!
!!! WJ-8611        !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRx1.driver:             C:\E3238s\lib\HD_8611.dll
!handoffRx1.interface:          rs232,ASRL0,9600
!handoffRx1.label:              VHF/UHF  Rx
!
!!!!!!!!!!!!!!!!!!!!!
!!! WJ-8615P       !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRx1.driver:             C:\E3238s\lib\HD_8615P.dll
!handoffRx1.interface:          gpib,hpib,5
!handoffRx1.label:              VHF/UHF  Rx
!
!!!!!!!!!!!!!!!!!!!!!
!!! WJ-8629        !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRx1.driver:             C:\E3238s\lib\HD_8629.dll
!handoffRx1.interface:          vxi,vxi,1
!handoffRx1.label:              VHF/UHF  Rx
!
!!!!!!!!!!!!!!!!!!!!!
!!! WJ-8629A       !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRx1.driver:             C:\E3238s\lib\HD_8629A.dll
!handoffRx1.interface:          vxi,vxi,1
!handoffRx1.label:              VHF/UHF  Rx
!
!!!!!!!!!!!!!!!!!!!!!
!!! WJ-8633-1      !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRx1.driver:             C:\e3238s\lib\HD_8633_1.dll
!handoffRx1.interface:            vxi,vxi,1
!handoffRx1.label:              VHF/UHF  Rx
!
!
```

```
!!!!!!!!!!!!!!!!!!!!!
!!! WJ-8711        !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRx1.driver:             C:\E3238s\lib\HD_8711.dll
!handoffRx1.interface:          rs232,ASRL0,9600
!handoffRx1.label:              HF  Rx
!
!!!!!!!!!!!!!!!!!!!!!
!!! WJ-8712        !!!
!!!!!!!!!!!!!!!!!!!!!
!   Note: Use driver HD_8712L.dll if the 8712 does
!         NOT have the extended IF bandwidths option.
!handoffRx1.driver:             C:\E3238s\lib\HD_8712L.dll
!handoffRx1.driver:             C:\E3238s\lib\HD_8712.dll
!handoffRx1.interface:          rs232,ASRL0,9600
!handoffRx1.label:              HF  Rx
!
!!!!!!!!!!!!!!!!!!!!!
!!! WJ-8721        !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRx1.driver:             C:\E3238s\lib\HD_8721.dll
!handoffRx1.interface:          vxi,vxi,24
!handoffRx1.label:              HF  Rx
!
!!!!!!!!!!!!!!!!!!!!!
!!! WJ-8723        !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRx1.driver:             C:\E3238s\lib\HD_8723.dll
!handoffRx1.interface:          rs232,ASRL0,9600
!handoffRx1.label:              HF  Rx
!
!!!!!!!!!!!!!!!!!!!!!
!!! WJ-8634        !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRx1.driver:             C:\E3238s\lib\HD_8634.dll
!handoffRx1.interface:          vxi,vxi,25
!handoffRx1.label:              VXI UHF/VHF Rx
!
!!!!!!!!!!!!!!!!!!!!!
!!! Cubic VXI-3250 !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRx1.driver:             C:\E3238s\lib\HD_3250.dll
!handoffRx1.interface:          vxi,vxi,26
!handoffRx1.label:              HF  Rx
!
!!!!!!!!!!!!!!!!!!!!!
!!! Cubic VXI-3550 !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRx1.driver:             C:\E3238s\lib\HD_3550.dll
!handoffRx1.interface:          vxi,vxi,26
!handoffRx1.label:              VHF/UHF  Rx
!
!!!!!!!!!!!!!!!!!!!!!
!!! Cubic VXI-3570 !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRx1.driver:             C:\E3238s\lib\HD_3570.dll
!handoffRx1.interface:          vxi,vxi,26
!handoffRx1.label:              VHF/UHF  Rx
!
!!!!!!!!!!!!!!!!!!!!!
!!! Cubic R-2411/U !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRx1.driver:             C:\E3238s\lib\HD_2411.dll
!handoffRx1.interface:          gpib,hpib,6
!handoffRx1.label:              MF/HF RCVR
!
!!!!!!!!!!!!!!!!!!!!!
!!! Cubic R-2412/U !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRx1.driver:             C:\E3238s\lib\HD_2412.dll
!handoffRx1.interface:          gpib,hpib,5
!handoffRx1.label:              VHF/UHF RCVR
!
!!!!!!!!!!!!!!!!!!!!!
!!! ICOM R20       !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRxN.driver:    C:\E3238s\lib\HD_icom_r20.dll
!handoffRxN.interface: rs232,ASRL<N>,1200,4a,0
!                           |       |     | | |
!                           |       |     | | +-- non-zero for debug output
!                           |       |     |  +-- radio address in hex
!                           |       |     +-- baud rate
!                           |       +-- ASLR1 or ASRL2 (VISA names for COM1..)
!                           +-- must be rs232
```

```
!
!handoffRx1.driver:    C:\E3238s\lib\HD_icom_r20.dll
!handoffRx1.interface: rs232,ASRL2,9600,6c,0
!handoffRx1.label:     VHF/UHF RCVR
!
!!!!!!!!!!!!!!!!!!!!!
!!! ICOM R7000     !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRxN.driver:    C:\E3238s\lib\HD_icom_r7000.dll
!handoffRxN.interface: rs232,ASRL<N>,1200,4a,0
!                           |       |      | | |
!                           |       |      | | +-- non-zero for debug output
!                           |       |      |  +-- radio address in hex
!                           |       |      +-- baud rate
!                           |       +-- ASLR1 or ASRL2 (VISA names for COM1..)
!                           +-- must be rs232
!
!handoffRx1.driver:    C:\E3238s\lib\HD_icom_r7000.dll
!handoffRx1.interface: rs232,ASRL2,9600,6c,0
!handoffRx1.label:     VHF/UHF RCVR
!
!!!!!!!!!!!!!!!!!!!!!
!!! ICOM PCR1000   !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRxN.driver:    C:\E3238s\lib\HD_IcomPcr1000.dll
!handoffRxN.interface: rs232,ASRL<N>,9600,0
!                           |       |      | |
!                           |       |      | | +-- non-zero for debug output
!                           |       |      +-- baud rate (must be 9600)
!                           |       +-- ASLR1 or ASRL2 (VISA names for COM1..)
!                           +-- must be rs232
!
!handoffRx1.driver:    C:\E3238s\lib\HD_IcomPcr1000.dll
!handoffRx1.interface: rs232,ASRL2,9600,0
!handoffRx1.label:     VHF/UHF RCVR
!
!
!!!!!!!!!!!!!!!!!!!!!
!!! ICOM R8500     !!!
!!!!!!!!!!!!!!!!!!!!!
! handoffRxN.driver:    C:\E3238s\lib\HD_icom_r8500.dll
! handoffRxN.interface: rs232,ASRL<N>,1200,4a,0
!                           |       |      | | |
!                           |       |      | | +-- non-zero for debug output
!                           |       |      |  +-- radio address in hex
!                           |       |      +-- baud rate
!                           |       +-- ASLR1 or ASRL2 (VISA names for COM1..)
!                           +-- must be rs232
!
!handoffRx1.driver:    C:\E3238s\lib\HD_icom_r8500.dll
!handoffRx1.interface: rs232,ASRL2,9600,6c,0
!handoffRx1.label:     VHF/UHF RCVR

!!!!!!!!!!!!!!!!!!!!!
!!! TenTec RX-331  !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRxN.driver:    C:\E3238s\lib\HD_TenTecRx331.dll
!handoffRxN.interface: rs232,ASRL<N>,9600,1,0
!                           |       |      | | |
!                           |       |      | | +-- non-zero for debug output
!                           |       |      |  +-- radio address
!                           |       |     +-- baud rate
!                           |       +-- ASLR1 or ASRL2 (VISA names for COM1..)
!                           +-- must be rs232
!
!handoffRx1.driver:    C:\E3238s\lib\HD_TenTecRx331.dll
!handoffRx1.interface: rs232,ASRL2,9600,1, 0
!handoffRx1.label:     HF RCVR
!
!!!!!!!!!!!!!!!!!!!!!
!!! Software       !!!
!!!!!!!!!!!!!!!!!!!!!
!handoffRx1.driver:             C:\E3238s\lib\HD_software.dll
!handoffRx1.label:              Software Rx
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Agilent RF Sensor       !!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! The Agilent RF Sensor software must be installed to use this driver.
!handoffRx1.driver:     C:\Program Files\Agilent\RFSensor\bin\HD_sensor.dll
!handoffRx1.interface:  sensorName
!handoffRx1.label:      sensorName
!
!
```

```
!!!!!!!!!!!!!!!!!!!!!!
!!! Agilent 89441A !!!
!!!!!!!!!!!!!!!!!!!!!!
!handoffRx1.driver:             C:\E3238s\lib\HD_89441.dll
!handoffRx1.interface:          telnet,11.22.33.44
!handoffRx1.interface:          gpib,hpib,19
!handoffRx1.label:              Signal Analyzer
!
!!!!!!!!!!!!!!!!!!!!!!
!!! Agilent 89600  !!!
!!!!!!!!!!!!!!!!!!!!!!
! See C:\E3238s\89600\Readme.txt for instructions
!handoffRx1.driver:             C:\E3238s\lib\hd_89600.dll
!handoffRx1.interface:          hostname, 7016
!handoffRx1.label:              Signal Analyzer
!
!!!!!!!!!!!!!!!!!!!!!!
!!! WiNRADiO
!!!!!!!!!!!!!!!!!!!!!!
!  "interface" should be "application,<SN>"
!  where <SN> is optional. Use it to select from multiple receivers.
!  Start the WiNRADiO windows application before starting E3238S.
!
!handoffRx1.driver:             C:\e3238s\lib\\HD_WinRadio.dll
!handoffRx1.interface:          application
!handoffRx1.label:              WiNRADiO Rx
!

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                   Menu Access Configuration                               !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! You can disable access to any of the features that are listed in the
! main menus (those that appear in the menu bar).  Using the 'disableAccess'
! command in this configuration file sets the initial status to disabled.
! Access to these features can be enabled while the application is running
! from the File, Access Control ... dialog box.  This is password protected
! if the user name "e3238s" appears in the password file.  If no such entry
! exists, there are no restrictions on menus access that a user can't change.
!
! To disable a particular menu, use the 'disableAccess' command with an
! argument that is a string that exactly matches the feature label.  You can
! also check the Access Control dialog box for the button label strings.
! Any feature not specifically disabled is, be default, enabled at startup.
!
!disableAccess: Secure Display
!disableAccess: Log Files ...
!disableAccess: Clear Log
!disableAccess: Clear Log File
!disableAccess: Clear Energy History
!disableAccess: Clear Signal Database
!disableAccess: Clear Frequency Lists
!disableAccess: Clear Audio Output
!disableAccess: Clear All
!

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                   Socket Server Configuration                             !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! The E3238S provides communication to the external world via sockets.
! To enable the socket interface on the workstation you must create a port
! number and enable the E3238S socket commands.  To create the port number
! enter a line in the C:\WINNT\system32\drivers\etc\services file as follows:
!          e3238s    7011/tcp
!
! The socket configuration is defined with the following commands:
!
!
! Parameter/command           Range                   Default value
! ------------------          ------------            --------------
! socketServer:               {Disabled or Enabled}   Disabled
! maxServices:                <1 to 5>                1
! maxClientSockets:           <0 to 16>               0
! socketServerTimerInterval:  {1 to 10000 mSec}       5
!
socketServer:               Disabled
maxServices:                1
maxClientSockets:           0
socketServerTimerInterval:  5

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                 E3238s Service Configuration                              !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
! The E3238s service is defined with the following commands:
```

```
!
! Parameter/command            Range                    Default value
! ------------------           ------------             --------------
! e3238sService:               {80 character string}    e3238s
! e3238sServiceDataBufferSize: {512 to 4194304}         512
! e3238sServiceMaxConnections: {1 to 32}                4
! e3238sServiceSendBufferSize: {0 to 8388608}           0 (Use system default)
! e3238sServiceRecvBufferSize: {0 to 8388608}           0 (Use system default)
!
e3238sService:               e3238s
e3238sServiceMaxConnections: 4
e3238sServiceDataBufferSize: 512
e3238sServiceSendBufferSize: 0
e3238sServiceRecvBufferSize: 0
!

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                    User Programming Shared Libraries                        !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Option ASD also provides the ability for users to implement new
! capabilities and add them to the E3238S application by creating
! shared-library programs.  The documentation supporting writing such
! programs appears in the ASD Programming Guide under "Adding Functionality
! to the E3238S".  See also, the Configuration section of the Command Ref.
!
! The libraries for each of the user-defined feature sets are loaded with
! the following commands:
!
userThreshold:
energyHistoryFilter:
userAlarmTask:
featureExtraction:
userMenu:
userPane:
genericLib:
!

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                          Signal Processing                                  !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!multipleSignalsPerProcessor: Enabled

!asxDsp_0:   C:\E3238s\downloadables\ASXdsp_0.elf
!asxDsp_1:   C:\E3238s\downloadables\ASXdsp_1.elf

!searchRx1.minDelayTimeRequired: 0

!
! Demo Signal Type
!

!signal1.enabled:     False
!signal1.hostLib:     C:\E3238s\demo\demoHost.dll
!signal1.hostDsp:     C:\E3238s\demo\demoDsp.dll
!signal1.targetDsp:   C:\E3238s\demo\demoDsp.esl
!signal1.loadFactor:  64
!signal1.minChannels: 4
!signal1.maxChannels: 32
!signal1.args:
!signal1.alias:

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                        Direction Finding                                    !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!df1.enabled:         False
!df1.hostLib:
!df1.args:
!df1.alias:
!df1.latitude:
!df1.longitude:
!df1.declination:
!df1.heading:


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!                      Modulation Recognition                                 !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!modRec1.enabled:       False
!modRec1.hostLib:       C:\E3238s\MR1\MR1.dll
!modRec1.args:
!modRec1.alias:
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!               Calibration Corrections Configuration                  !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! These commands control the application of corrections to compensate
! for tuner response and antenna path response. These user-supplied
! corrections are in addition to built-in RF and IF corrections.
! For an example of the file format, see the file e3238s\cal\d.tuner1.cal.
!
!     searchRx<N>.tuner<M>.userCalFile:  <filename>
!     searchRx<N>.antenna<M>.calFile:    <filename>
!
! Cal files are normally located in the C:\E3238s\cal  directory.
!
!searchRx1.tuner1.userCalFile:  tuner1.cal
!searchRx1.antenna1.calFile:    antenna1.cal
```