

---

# Evaluating System Distortion Effects by Simulation or Measurement Methods

## Product Note E4600-6

---

### Overview

The Series IV Communications Design Suite (CDS) and OmniSys software include the capability to share data with the HP 89400A family of Vector Signal Analyzers (VSAs). Communications system designers can combine actual hardware measurements with software simulation data. The power of software simulation and flexibility of the HP 89400A VSAs allow complete systems to be assembled and tested even when some hardware subsystems have not been completed.

Beginning with Version 6.0, OmniSys/CDS can read measurement data from the VSA, use that data in a simulation, and send the simulation results back to the VSA. Using its arbitrary signal generation ability, the VSA can then use the simulated data to create a real signal to test other hardware subsystems.

Some applications of OmniSys/CDS and a VSA include:

- A subsystem can be tested by emulating missing components with OmniSys/CDS and a VSA.
- A new modulation scheme or constellation shape can be derived from a VSA arbitrary source when it is controlled by OmniSys/CDS.
- A channel fade, interference, or diversity scenario can be emulated with OmniSys/CDS propagation models and placed in the actual transmission path by use of a VSA.

This product note provides VSA and workstation configuration information to accomplish two-way data transfer. A demonstration of an interconnected system of VSA and workstation, including an amplifier in compression showing both simulated and measured distortion data, is presented.

### Introduction

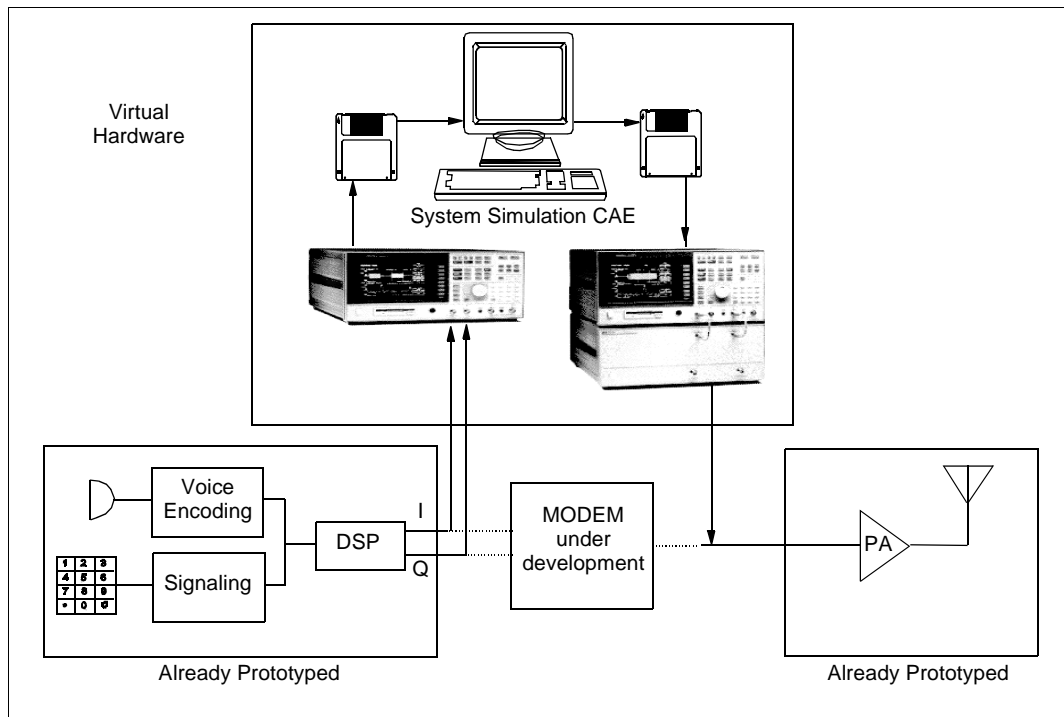
As Figure 1 illustrates, OmniSys/CDS can be used in tandem with a VSA to allow an entire system to be tested even though part of the system has not been completed. The *MODEM under development* can be bridged using a VSA and OmniSys/CDS with the virtual hardware concept. A prototyped DSP voice encoder output can be recorded by a VSA. VSA analysis data can then be used to drive an OmniSys/CDS simulation of the *MODEM under development*. Simulation output can then be used to provide input data

for a VSA arbitrary source, which can then be used to electrically drive the prototyped power amplifier/antenna.

Using file transfer methods, the VSA and any simulated sub-system becomes virtual hardware that allows:

- system measurement results to be included in the system simulation; in Figure 1, see the left and center blocks.
- pre-prototype sub-system emulation; in Figure 1, see the virtual hardware block in the center on the workstation bridging the modem.
- arbitrary VSA signal generation for RF sub-system stimulation; in Figure 1, see the center and right blocks.

In this way, hardware effects can be included in a simulation, or simulation effects can be included in the hardware.



**Figure 1. OmniSys/CDS used in tandem with a VSA**

### VSA Data Transfer and Control

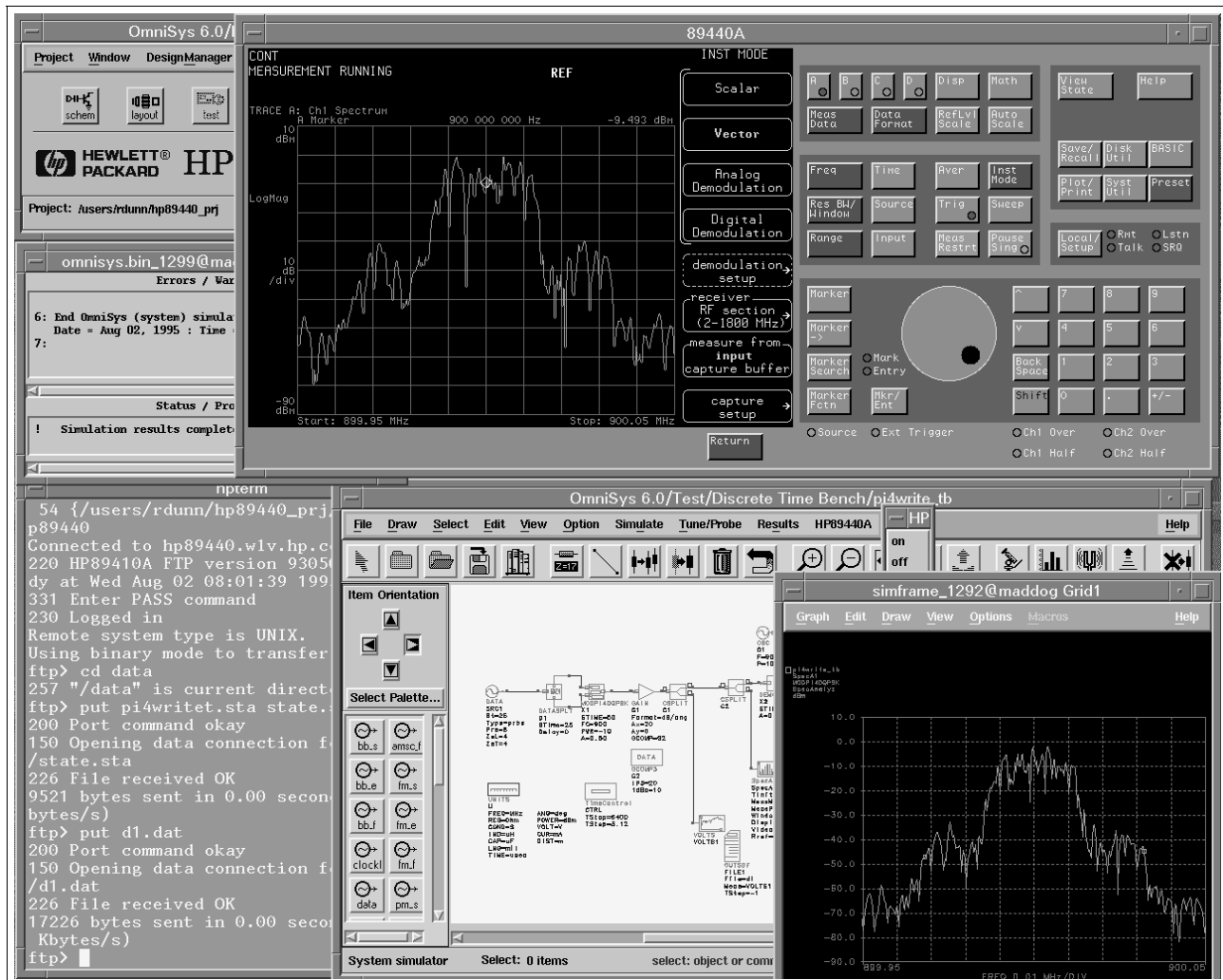
Data can be transferred between a computer and a VSA via either floppy disk exchange or file transfer protocol (*ftp*) on a local area network (LAN). Besides data transfer, a VSA can be remotely controlled through an HP-IB or LAN (along with normal front panel control). Data is *transferred* between the computer and a LAN-connected VSA using *ftp*, regardless of how the VSA is being controlled.

If the VSA is connected to a controlling computer via HP-IB, the VSA control language Standard Commands for Programmable Instruments (SCPI) can be sent from the computer to the VSA using HP BASIC commands.

If the VSA is connected to a LAN, it can be remotely controlled from the workstation in either of two ways:

- If a *telnet* session is established between the computer and the VSA, SCPI language can be sent without the use of the HP-IB or HP BASIC. Data is transferred by means of a concurrent *ftp* session.
- If an X-Window is opened by the VSA in a *telnet* session, an image of the VSA front panel can be displayed along with OmniSys/CDS (see Figure 2). This image is fully functional, in that all VSA front-panel controls, soft- and hard-keys can be operated with the remote computer mouse or keyboard (or both). The same *telnet* session used to open the X-Window can also be used to send SCPI control language, if desired. Data is transferred by means of a concurrent *ftp* session. The VSA becomes a virtual instrument.

Details on these operating modes are presented in this product note.



**Figure 2.** Remote display of the VSA front panel can be established as shown in the upper-right portion of the screen. The VSA can be controlled by means of a user-defined menu in the OmniSys/CDS test bench, or by a telnet/SCPI session (not shown), or both. An OmniSys/CDS session can be run simultaneously with the remote operation of the VSA as shown in the upper left and lower right corners. A simultaneous *ftp* session is shown in the lower left corner.

## OmniSys/CDS Features

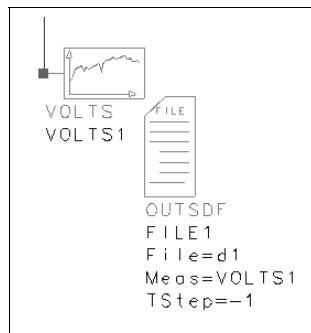
The file format common to OmniSys/CDS and the VSA is called SDF (Standard Data Format). SDF files are binary, have *.dat* extensions, and are written into, and read out of the data sub-directory of the current project.

In OmniSys/CDS, the OUTSDF item can write SDF files; the QAM\_F item can read SDF files. Both the OUTSDF and QAM\_F items can be used in either the Signal FFT Test Bench or the Discrete Time Test Bench—the Discrete Time Test Bench is used in this product note. The application details of OUTSDF and QAM\_F are different in the Signal FFT Test Bench. (Refer to the Series IV System Test Items manual for details regarding these items and test benches.)

The OUTSDF item creates a complex, binary, time-domain data file in SDF format based on the VOLTS measurement. (Other measurements can also be referenced with the OUTSDF item but they are not for complex IQ measurements.) This measurement includes noise embedded in the signal such as from baseband noise generators or as a result of having *ENoise=On* in the TimeControl item.

The VOLTS measurement passes complex, time-sampled data based on the TStep parameter with the *Fc* (carrier frequency in use at the VOLTS measurement point) to the OUTSDF item, which then creates an SDF file. If TStep= 1, the time step size of the signal at the measurement point will be used; if re-sampling is desired, set TStep to the desired value of sampling time. When specifying a file name in the OUTSDF Item Parameters window, note that the *Value Options* button allows access to a list of all *.dat* files in the data sub-directory. A new, unique name can be typed directly into the *Value* box. Multiple OUTSDF items must have unique file names.

Figure 3 shows the OUTSDF item as it appears in the test bench.



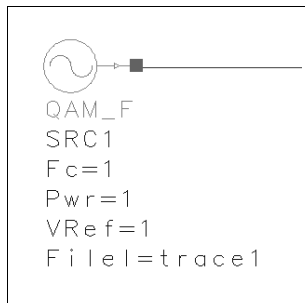
**Figure 3. OUTSDF item**

The QAM\_F item generates an OmniSys/CDS quadrature amplitude modulated signal whose I and Q signals are based on a complex, time-domain SDF file. Embedded noise characteristics in the VSA signal are preserved in the QAM\_F signal. OmniSys/CDS signal characteristics such as *Fc*, *Pwr*, *Vref*, and *Rref* (*Zgen*), are set by the file contents. While the three explicit QAM\_F parameters are ignored, they must be given default values to satisfy the source syntax; *Fc=Pwr=Vref=1* are acceptable defaults. In addition, the file name need be entered only once under FileI, while FileQ can be left blank or \* entered. Use of \* suppresses the appearance of FileQ= in the Schematic or Test window. The same name can also be entered under FileQ, but it is not necessary. (This is not true for non-SDF files; other formats can have different syntax. Refer to the System Test Items manual for details.) The *Value Options* button in the QAM\_F Item Parameters window allows access to a list of all *.dat* files in the data sub-directory.

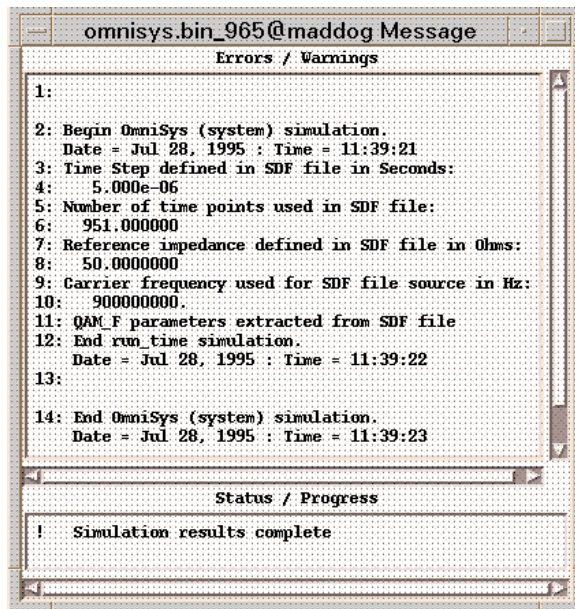
Figure 4 shows the QAM\_F item as it appears in the test bench.

Each time OmniSys/CDS reads an SDF file with the QAM\_F source, SDF file header information is displayed in the simulator dialog box; see Figure 5.

Header information is displayed in the simulator dialog box so that appropriate settings can be made in the simulation, such as TimeControl step size and other related items. In addition to simulation-specific information, the version number and domain (time or frequency) are also read from the file header. While this information is not displayed in the simulator dialog box, OmniSys/CDS uses this information to establish compatibility with the current SDF file. OmniSys/CDS and current versions of the VSA write Version 3 files. In addition, OmniSys/CDS reads and writes SDF Version 3 time domain files only, while the VSA reads Versions 2 and 3 time and frequency domain files.



**Figure 4. QAM\_F item**



**Figure 5. Typical simulator dialog box display of OmniSys/CDS reading an SDF file while using a QAM\_F source**

**Note** There are certain demonstration files supplied by HP that use SDF Version 2. These files can be updated to Version 3 with the utility *sdf2osdf* available as a DOS executable which is shipped with the VSA manual set. This utility is also available for HP-UX on HP Series 700 workstations.

This DOS or HP-UX utility, as well as six other utilities, are also available for *anonymous ftp* downloading on [hpeesof.external.hp.com/distribution/app\\_note/sdfutil](http://hpeesof.external.hp.com/distribution/app_note/sdfutil). These utilities are also available on World-Wide Web at <http://www.hp.com/go/hpeesof>.

A partial listing of an SDF file header is shown next (the actual header is very lengthy).

```
TRACE1.TXT

File 'sdfdat\tracel.dat, size 9690 bytes
.....
FORMAT_STRUCT
format:                                "B"
.....
SDF_FILE_HDR
...
revisionNum:                           3
...
.....
SDF_MEAS_HDR
...
centerFreq:                             900000000
...
.....
SDF_DATA_HDR 0
...
domain:                                 TIME_DOMAIN
...
yIsComplex                               1
...
xUnit.factor:                           1
...
yUnit.factor:                           1
...
abscissa_deltaX:                        5e-006
...
num_of_points:                          951
...
.....
SDF_CHANNEL_HDR 0
...
inputImpedance:                         50
...
.....
```

## OmniSys/CDS Use of Power and Frequency Information from SDF Files

When OmniSys/CDS reads a VSA file with a QAM\_F source, all of the simulation source parameters are controlled by the contents of the file. The Fc value specified in the file, which was the VSA measurement receiver *center* frequency, automatically becomes the simulation *carrier* frequency (also called Fc). Since the Fc parameter on the QAM\_F item is ignored, the simulation element, Signal IQ Characterization Frequency Change (IQFC) can then be used to reassign the simulation Fc, if necessary. (Refer to the System Network Items manual for details regarding the IQFC element.)

There is no explicit power level setting in the SDF file itself. However, the VSA measurement receiver output files (*traceN.dat*) have voltage levels set such that associated powers are always leveled to +10 dBm (0 dBVpk in a 50-ohm system). OmniSys/CDS then automatically sets the carrier level in the QAM\_F source to + 10dBm. This level is unaffected by the *Pwr* and *VRef* parameter settings on the QAM\_F source.

## VSA Use of Power and Frequency Information from SDF Files

When the VSA reads an OmniSys/CDS file that uses an OUTSDF item, the file header Fc is not used to set the measurement receiver center frequency. The VSA architecture is such that the source frequency always tracks the measurement receiver center frequency. The source has a maximum of 4096 complex data/time points. The measurement receiver center frequency can be changed with the *Freq / center* keys; the source center frequency is automatically changed also. When a *.dat* file is read in by the VSA source with a new frequency, the VSA message *Arbitrary Source data center changed* means the new frequency request is ignored and the old receiver frequency is still in use. In other words, the source center (frequency) is changed from the new to the old frequency.

The power level of the VSA source is not controlled by the SDF file. Rather, it is set by the soft key under *Source / level*. The file I and Q voltage levels are then normalized to appropriate levels for the soft key power level. An SCPI command can also be used to set the source level in a *telnet* session or by HP-IB control. In this case, the *sour:volt 0 dbvpk* command as well as the *sour:volt 10 dbm* can be used. (Note that dBm units are not available on the VSA front panel soft keys.)

## VSA Description

The HP 89400A VSA family has a common measurement engine, the HP 89410A dc-to-10-MHz baseband analyzer. The HP 89410A acts as the user interface, final digital IF section, signal processor, source, and display section for the entire family.

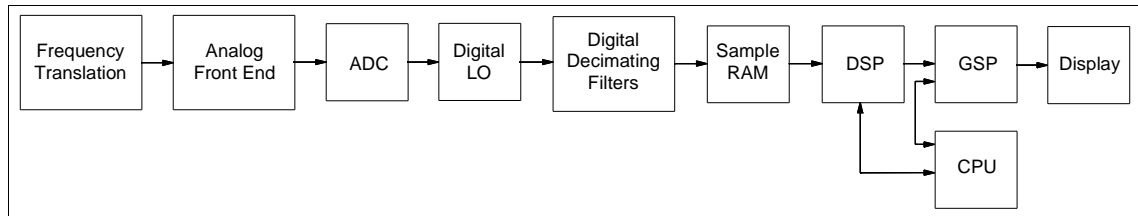
The HP 89440A, HP 89441A, and HP 89411A use analog RF hardware to down-convert higher-frequency bands into a 7-MHz information bandwidth centered at 6 MHz. This provides measurement capabilities at RF employing baseband DSP techniques.

The HP 89410A also includes a dc-to-10-MHz signal source that can read in a maximum of 4096 complex time points. Besides CW, random noise, and periodic chirp, the source can be driven from an arbitrary waveform file. Refer to the previous section *VSA Use of Power and Frequency Information from SDF Files*.

The HP 89440A and HP 89441A also up-convert the source to the same band as the receive section. The HP 89411A is a down-converter that accepts a 21.4 MHz IF output from a microwave spectrum analyzer.

The HP 89440A extends the range of both measurement receiver and source from 2 MHz to 1.8 GHz; the HP 89441A extends the range to 2.65 GHz.

The high-level measurement processing used in the VSA is shown in Figure 6. The HP 89410A has an alias-protected analog input bandwidth of dc to 10 MHz. The analog input signal flows into a hybrid dithered ADC, which samples the signal at a rate of 25.6 MHz. The digital data stream from the ADC is routed to the digital LO, which performs frequency translation and quadrature detects the digitized input signal, resulting in a complex data stream with real and imaginary parts. This full-rate data is then input to the digital decimating filters. These filters perform binary decimation and provide image rejection. The output of the digital filters represents a band limited digital version of the analog input signal in the time domain. This digital data stream is then captured in a sample RAM. This sample RAM is a circular FIFO buffer that collects individual data samples into blocks to be manipulated by the DSP.



**Figure 6. HP 89400A Signal Flow Diagram**

The fundamental data used to produce all measurement results, whether in the frequency or time domain, is the time-domain data collected in the sample RAM. This can be contrasted with a swept spectrum analyzer, which produces results directly in the frequency domain using swept filter and detection techniques. Since the input signal is processed and captured in this fundamental time-domain form, all attributes of the signal at that moment in time are represented: amplitude, frequency, and phase.

The HP 89400A uses a set of three processors to perform the measurement post-processing on the sample RAM time data. These processors are under the control of a real-time multi-tasking operating system. The main CPU, which handles system management and processes user input, is a Motorola 68EC030. The digital signal processor is the Motorola DSP96002 floating-point dual-port processor. This processor performs all block-oriented mathematical operations required by the DSP algorithms. The Texas Instruments TMS34020 graphics signal processor is used. A data throughput rate of 60 updates per second can be supported.

The overall design of the VSA is based on an FFT. This architectural choice is made because of the improved speed and information processing capability when compared to a conventionally swept spectrum analyzer. In a swept analyzer, its (one) filter must be swept to a given frequency and its output settled before a result can be obtained at that frequency. The filter is then swept to the next frequency, and so on.

The speed improvement of an FFT algorithm occurs because a bank of parallel filters is emulated such that they can be simultaneously settled and outputs obtained at all frequencies concurrently. In this way, for FFT filter resolution comparable to the swept analyzer, the FFT measurement can be much faster. In addition to a speed advantage, information processing capability is improved with the FFT technique. In a swept analyzer, phase information is difficult to obtain and transient signals are difficult to capture.

Since the time domain is the fundamental data type, all characteristics of the signal taken by the FFT measurement are preserved for subsequent analysis. The FFT algorithm is



useful for characterization of amplitude and phase at all frequencies simultaneously. In addition, the FFT technique is useful for measuring time-variant or transient signals.

Information reported under *VSA Description* is taken from several articles published in the **Hewlett-Packard Journal**, December 1993, Volume 44, Number 6.

## VSA Remote Control

Programming the VSA directly replaces instrument operation from the front panel with commands sent by a computer over the HP-IB or LAN. The VSA can be remotely controlled using a set of SCPI commands. For example, instead of using a front-panel key to reset the analyzer, the SCPI \*RST command can be sent. In addition, there are functions that are not available from the front panel that can be accomplished via LAN such as electronic data transfer to or from the VSA.

A *telnet* session over a LAN can accomplish the same control as an HP-IB session. Once a *telnet* session is established for remote control, an image of the VSA front panel can also be displayed in an X-Window session on the host computer screen (see Figure 2). (While an X-Window can be displayed over a LAN through a *telnet* session, the HP-IB bus cannot support an X-Window.)

Once an X-Window is displayed, control can then be achieved by pushing the virtual VSA display buttons with the mouse or by SCPI command line arguments in the *telnet* session (or both); refer to Table 1. In addition, an *ftp* session can be simultaneously established with the *telnet* session so that data transfer between the workstation and VSA can also be accomplished. Multiple, concurrent *telnet* and *ftp* sessions from the same or different hosts can be supported by the VSA.

**Table 1. VSA Hard- and Soft-Key Equivalent SCPI Commands**

VSA hard and soft keys	SCPI commands (see Note)
Local / Setup / LAN setup / X11 display / on off	<code>syst:comm:lan:xwin[:stat?   on   off]</code>
Local / Setup / LAN setup / X11 IP addr xx.xx.xx.xx	<code>syst:comm:lan:xwin[:host?   :host 'xx.xx.xx.xx']</code>
Local / Setup / SCPI cmd echo	<code>syst:gpib:echo [ on   off ]</code>
Note: the symbols [   ] are not SCPI, they are used for delimiters.	

## VSA Hardware Configuration

In order to use the VSA for *telnet* and *ftp* sessions, UFG and UG7 options must be present.

The UFG option provides *4MB Extended RAM and Additional I/O (support)*, which includes the memory, a LAN interface containing a BNC ThinLAN (50-ohm coaxial cable) connector as well as AUI (15-pin) connector, an additional HP-IB port, and *telnet* capability.

The UG7 option provides *Advanced LAN Support*, which includes remote X-Window and *ftp* capabilities. ThinLAN 10Base-2 or AUI port types can be used, but not simultaneously. If 10Base-T (four twisted pair) is desired, an appropriate EitherTwist medium attachment unit (MAU) transceiver must be installed on the AUI port.

## VSA Formal Network Information

If the VSA is connected to a formal network, obtain an IP address, system name, gateway IP, and subnet mask. Even though the IP address (not the system name) is used by the VSA, a descriptive name such as *hp89440.your.company.com* is easy to remember when opening *telnet* or *ftp* sessions from a workstation. If no gateway or subnet mask is assigned, use *0.0.0.0*.

## VSA Standalone Network Information

If the VSA and workstation are operated standalone, the IP address must be picked to be on the same subnet as the workstation—the first three fields of the IP address must be identical and a name must be chosen. For instance, the VSA IP address could be one digit different from the workstation IP address. The VSA's new name and IP address must be entered into the workstation */etc/hosts* file.

If coax cable (10Base-2) is used to connect the VSA and workstation, tees with 50-ohm loads must be used on both ends.

If EitherTwist (10Base-T) is used, a hub is normally used for VSA and workstation connection in an office or lab environment. (The VSA and workstation cannot be connected directly with a standard EitherTwist cable.)

The VSA and workstation can be directly connected without the use of a hub if a *swapped* cable is used. Swapped cables are not normally used in office or lab environments, however they are used in rack and panel distribution installations. A *swapped* cable has the transmit and receive pairs interchanged: 1-3, 2-6, 3-1, 4-4, 5-5, 6-2, 7-7, 8-8. They are commercially available or can be locally built.

## VSA Configuration Data Entry

Once the VSA hardware is configured, the port select, IP address, gateway IP, and subnet mask are entered in the VSA *Local / Setup / LAN setup / LAN port setup* menu. The LAN power-on toggle choice must be made *active* in the *LAN setup* menu to turn on LAN functionality. Once all choices are entered in the *LAN port setup* menu, cycle power to the VSA IF section to allow the changes to become effective and permanent. It is not necessary to cycle power to the VSA for any other menu choices to become effective.

## Establishing a *telnet* Session for a SCPI Session and Remote VSA Display

The first time a *telnet* session is attempted on a new LAN configuration, ping the VSA from the workstation to establish that there is communication. (Ping can be performed with either workstation name or IP address used as an argument.)

```
{/users/rdunn/hp89440_prj}: /etc/ping hp89440.wlv.hp.com
```

The answer should come back:

```
PING hp89440.wlv.hp.com: 64 byte packets
64 bytes from 15.13.16.88: icmp_seq=0. time=14. ms
64 bytes from 15.13.16.88: icmp_seq=0. time=64. ms
---hp89440.wlv.hp.com PING Statistics---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip (ms) min/avg/max = 14/39/64
```

Ctrl-C can be used to interrupt the ping.

Once two-way communication is established, before attempting to open a remote X-Window, use the *xhost* command to allow the VSA to display a window on the host

workstation screen. *xhost +* must be used to allow any foreign workstation to open a remote window on a local host workstation. When a foreign workstation attempts to open a window on a local workstation screen without the use of *xhost*, nothing is displayed and no error message is returned. As shown below, *xhost* can be made specific by the use of either the remote host name or IP address as an argument. In the following example only *hp89440.wlv.hp.com* has access to open a window on the host workstation.

```
{/users/rdunn/hp89440_prj}: xhost +hp89440.wlv.hp.com
hp89440.wlv.hp.com being added to the access control list
```

A *telnet* session is then opened.

```
{/users/rdunn/hp89440_prj}: telnet hp89440.wlv.hp.com
Trying...
Connected to hp89440.wlv.hp.com.
Escape character is '^]'
Welcome to the SCPI Parser
```

A VSA *telnet* session does not require a user name or password. Once the session is open, *telnet* confirms the VSA connection and returns information that the SCPI session can be terminated or stopped with *Ctrl-J*. Control is then returned to the telnet command line. Following the line *Welcome to the SCPI Parser* the ready state for the SCPI command line is displayed. There is no other prompt or blinking cursor, only a blank line feed is visible. SCPI commands can then be entered; refer to Table 2 for a summary of SCPI commands.

To quit the SCPI parser and return control to telnet use *Ctrl-J*.

To return control to the workstation's UNIX command line, use *quit* at the *telnet* prompt. The UNIX *{/users/rdunn/hp89440\_prj}*: prompt line appears.

**Table 2. SCPI Command Summary\***

Action	SCPI Command
<code>syst:comm:lan:xwin:stat?</code>	Returns the current state of the X-Window: 1 for true (if it is on) and 0 for false (if it is off). Only one X-Window can be displayed on one workstation at a time.
<code>syst:comm:lan:xwin:host?</code>	Returns the IP address for the current X session host.
<code>syst:comm:lan:xwin:host '15.13.16.81'</code>	Assigns the X-Window host to be the stated IP address. The host address (not the name) must be used.
<code>syst:comm:lan:xwin on</code>	Toggles the remote X-Window display on. (The X-Window appears on the host named 15.13.16.81).
<code>syst:gpib:echo on</code>	Displays the current SCPI command in the upper left corner of the VSA display.
<code>syst:comm:lan:xwin off</code>	Toggles the remote X-Window display off.
* Complete SCPI command reference information can be found in HP 89400-Series HP-IB Command Reference, HP Part No. 89441-90027.	

## Typical SCPI Session

After a *telnet* session to the VSA is opened, a SCPI session can begin when *syst:comm:lan:xwin:stat?* is typed into the SCPI command line. (Table 2 is a summary of SCPI commands.) If +1 is returned, the VSA X-Window is being displayed on a workstation; if +0 is returned, the VSA X-Window is not being displayed on another workstation.

Because a VSA can support multiple, simultaneous, concurrent *telnet* (and *ftp*) sessions from multiple workstations, the fact that a *telnet* session can be opened to a VSA does not mean that this is the only *telnet* (or *ftp*) session with that VSA. To find out where the VSA X-Window is being displayed, use *syst:comm:lan:xwin:host?* A host is the workstation on which the remote X-Window is displayed. An IP address will be returned with which the UNIX command *nslookup* (available on the workstation) can be used to find the workstation name associated with that IP address. This name can then be used to determine who is using the VSA. Because there is no network security on a VSA, the command *syst:comm:lan:xwin off* could be issued to terminate the X-Window session on the other host system.

The desired X-Window host IP address is set on the VSA by using the SCPI command *syst:comm:lan:xwin:host '15.13.16.81'* where *15.13.16.81* is the host machine address. The *nslookup* command can also be used to find an IP address corresponding to the host name. The host name can be found with the UNIX command *hostname* (available on the workstation). The VSA X-Window IP address cannot be changed while an X-Window is being displayed. (*xhost +* must be used at the workstation UNIX command line prior to opening the remote VSA display.) Then, *syst:comm:lan:xwin on* will display the VSA front panel on the workstation screen. When the VSA front panel is displayed, any SCPI *error* messages will be shown in a box on the VSA display for approximately 5 seconds.

To find the SCPI command that corresponds to any front panel hard- or soft-key combination, turn on the SCPI command echo with *syst:gpib:echo on*. The VSA will then display the most recent SCPI command as a result of either key or command entry in the upper-left corner of the X-window. The displayed syntax can then be re-typed into the SCPI parser command line (in either upper or lower case), to achieve the same effect as hard or soft keys. Then *syst:comm:lan:xwin off* will close the remote display, *Ctrl-]* will close the SCPI session, and *quit* in the *telnet* session will return control to the workstation command line.

## VSA File System

The VSA user-accessible file system consists of (non-volatile) NVRAM, RAM, and a removable DOS or LIF format floppy disk. The floppy disk is known as an *internal* disk on the VSA. An optional HP-IB external hard disk can also be connected to the VSA. From the front panel *default disk* control, each of these appears as a disk in the VSA file system. During an *ftp* session, the first three disks appear as subdirectories in the VSA root file system, except the hard disk, which is not accessible in an *ftp* session.

When operating the VSA without a network, its file system can be used to read source files, write measurement receiver data, and save instrument states to or from the *internal* floppy disk. When the VSA is used in conjunction with OmniSys/CDS, files can be transferred by interchanging a floppy disk between the workstation and VSA, assuming the workstation can read DOS or LIF format.

## VSA Network Use

When the VSA is operated remotely on a LAN, an *ftp* session can be used to access and exchange VSA data with OmniSys/CDS. When the VSA is accessed in an *ftp* session, the file system is structured somewhat differently than when accessed from the front panel. The internal floppy disk appears as a subdirectory called *int*; nonvolatile RAM appears as a subdirectory called *nvr*; volatile RAM appears as a subdirectory called *ram*; and a new subdirectory called *data* is present.

*data* is a directory of files that are actually memory-mapped data locations that can be transferred between the VSA and OmniSys/CDS. Refer to Table 3 for the types of information in these files. Current data in *trace1.dat*, *trace2.dat*, *trace3.dat*, *trace4.dat* form the basis for VSA measurements A, B, C, and D, respectively. If the measurement is displayed in the time domain, so is its corresponding data; if the measurement is displayed in the frequency domain, so is its corresponding data. If measurements are based on a VSA Inst Mode / Digital Demodulation, time domain data is automatically corrected; if measurements are based on a VSA Inst Mode / Vector, time data is not corrected. The VSA Syst Util / time domain cal / **on** can then be applied to the data for correction. The current display is available in several screen capture formats as *screen.xxx*.

**Note** OmniSys/CDS cannot read frequency domain data; *traceN.dat* must be associated with a time domain measurement before transferring it to OmniSys/CDS.

**Table 3. *data* directory files**

VSA Memory Mapped Information	VSA Sub-directory / <i>data</i> File Name
Current Measurement Configuration	state.sta
Source Input Register 1	d1.dat
Source Input Register 2	d2.dat
.	.
.	.
.	.
Source Input Register 6	d6.dat
Analyzer Output Measurement Data Register A	trace1.dat
Analyzer Output Measurement Data Register B	trace2.dat
Analyzer Output Measurement Data Register C	trace3.dat
Analyzer Output Measurement Data Register D	trace4.dat
Screen Display Image in TIFF format	screen.tif
Screen Display Image in HP-GL format	screen.hgl
Screen Display Image in PCL format	screen.pcl
Screen Display Image in PGL format	screen.pgl

## VSA File Naming Conventions

Prior to beginning *ftp* sessions between a workstation and a VSA, establishing file naming conventions in OmniSys/CDS can simplify those sessions. It is also assumed that the *ftp* session originates in the OmniSys/CDS project *data* directory and targets the VSA *data* directory. By naming the OUTSDF file *d1.dat* and the QAM\_F file *trace1.dat* in OmniSys/CDS, file transfer then becomes *put d1.dat* or *get trace1.dat*. By using these (or similar) names, data exchange can be accomplished without renaming files. When a simulation or measurement update occurs, *put d1.dat* or *get trace1.dat* updates the appropriate VSA or workstation file, respectively.

Another convention is to keep the VSA *state.sta* file in the project data directory (rather than on the VSA), which then associates a specific VSA *state.sta* file with a particular project. Since the VSA itself cannot rename *state.sta* files, multiple file storage of these files can be accomplished by the use of source and target argument names with *get* or *put*. Thus, *state.sta* files can be renamed and saved for later use. For example, *get state.sta pi4meas.sta* transfers a copy of the VSA file named *state.sta* and renames it on the workstation to *pi4meas.sta*. Similarly, *put pi4meas.sta state.sta* transfers a copy of the workstation file from the VSA and renames it *state.sta*.

## Establishing a VSA *ftp* Session

Before attempting an *ftp* session, establish communication between the VSA and the workstation by either pinging the VSA from the workstation, or having a *telnet* session already established.

An *ftp* session can be opened from the project *data* sub-directory:

```
{/users/rdunn/hp89440_prj/data}: ftp hp89440.wlv.hp.com
Connected to hp89440.wlv.hp.com
220 HP89410A FTP version 930507 (KA9Q) ready at Fri Jul 14 16:20:40 1995
Name (hp89440:rdunn): rdunn
331 Enter PASS command
Password:
230 Logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Even though the VSA has no network security, the *ftp* protocol as implemented on the VSA requires a login name and a password. Any name and password will do, both must be used, and each must be at least one alphanumeric character (longer forms are used in this example file). Because SDF files are binary, the *ftp* session must be in binary mode—type *binary* at the *ftp* command line.

If a file exists in the workstation home directory called *.netrc*, has certain permissions, and contains machine, login, and password information, *ftp* will use the file as a lookup table to eliminate the need for command line entry of a login name and password each time a session is opened. This ASCII file must be composed as follows:

```
machine hp89440.wlv.hp.com
login rdunn
password demo123
```

The permissions must be: *-rwx----- (chmod 700 .netrc)*. Multiple entries of machine, login, and password names can appear in the *.netrc* file, if desired. This file is not required by the AEL script that creates a user-defined menu. The *.netrc* file is not used by *telnet*, as login names and passwords are not required by the VSA implementation.

## Typical VSA *ftp* Session

To list the VSA sub-directories:

```
ftp> ls
Port command okay
Opening data connection for LIST /
int          0      DIR 01-01-93  08:00:00 AM
nvram       0      DIR 01-01-93  08:00:00 AM
ram         0      DIR 01-01-93  08:00:00 AM
data        0      DIR 01-01-93  08:00:00 AM
```

In addition to *ls* used in this example, *dir*, *cd*, *lcd*, *pwd*, *!pwd*, and *!cd* also work. Relative paths can be used with *put* or *get*.

A partial listing of VSA files useful for exchange with OmniSys/CDS:

```
ftp> cd /data
ftp> ls
Port command okay
Opening data connection for LIST /data
d1.dat      Data register
...
d6.dat      Data register
trace1.dat  Trace measurement data (read only)
...
trace4.dat  Trace measurement data (read only)
state.sta   Instrument State
screen.tif  Screen image in TIFF format
```

### State Restoration and Data Transfer

The following *ftp* commands assume that the *ftp* source directory is a project *data* directory (that is, *ftp* started from there) and that the *ftp* target directory is a VSA *data* directory (that is, *ftp >cd /data*). Names of the OmniSys/CDS files are also assumed: *OUTSDF* file is *d1.dat*; *QAM\_F* file is *trace1.dat*. These *ftp* commands follow a login between a workstation and a VSA.

### Restoring a Previous Instrument Measurement State

The file *pi4write.sta* will be copied under the name *state.sta*. As soon as the state file is received, the VSA is restored to the previous state.

```
ftp> put pi4write.sta state.sta
Port command okay
Opening data connection for STOR /data/state.sta
File received OK
9594 bytes sent in 0.00 seconds (7055 Kbytes/s)
```

### Transferring Omnisys/CDS Data Files to the VSA

Each re-simulation of OmniSys/CDS causes a file named *d1.dat* to be updated in the project data directory (provided it is so named in *OUTSDF*). *put* transfers a copy of *d1.dat* from the workstation to the VSA following that simulation. When the VSA receives the file, the source is updated with current simulation data.

```
ftp> put d1.dat
Port command okay
Opening data connection for STOR /data/d1.dat
File received OK
17225 bytes sent in 0.00 seconds (9192 Kbytes/s)
```

### Transferring VSA Data Files to Omnisys/CDS

Once the VSA measurement has stabilized, *get* transfers a copy of *trace1.dat* from the VSA to the workstation. The *trace1.dat* file is then re-read into OmniSys/CDS upon re-simulation, which updates the current measurement data.

```
ftp> get trace1.dat
Port command okay
Opening data connection for RETR /data/trace1.dat
File sent OK
9690 bytes sent in 0.56 seconds (17.4 Kbytes/s)
```

Normal *ftp* session termination and return to the workstation command line:

```
ftp> bye
Goodbye!
{/users/rdunn/hp89440_prj/data}:
```

Table 4 is a summary of *ftp* commands.

**Table 4. *ftp* Command Summary**

Action	<i>ftp</i> Command
Restoring a VSA state	ftp> put pi4write.sta state.sta
Writing data out to the VSA source following an OmniSys/CDS simulation	ftp> put dl.dat
Reading data back from the VSA measurement to OmniSys/CDS prior to simulation	ftp> get trace1.dat
Saving a VSA state	ftp> get state.sta pi4write.sta

### Demonstration Setup

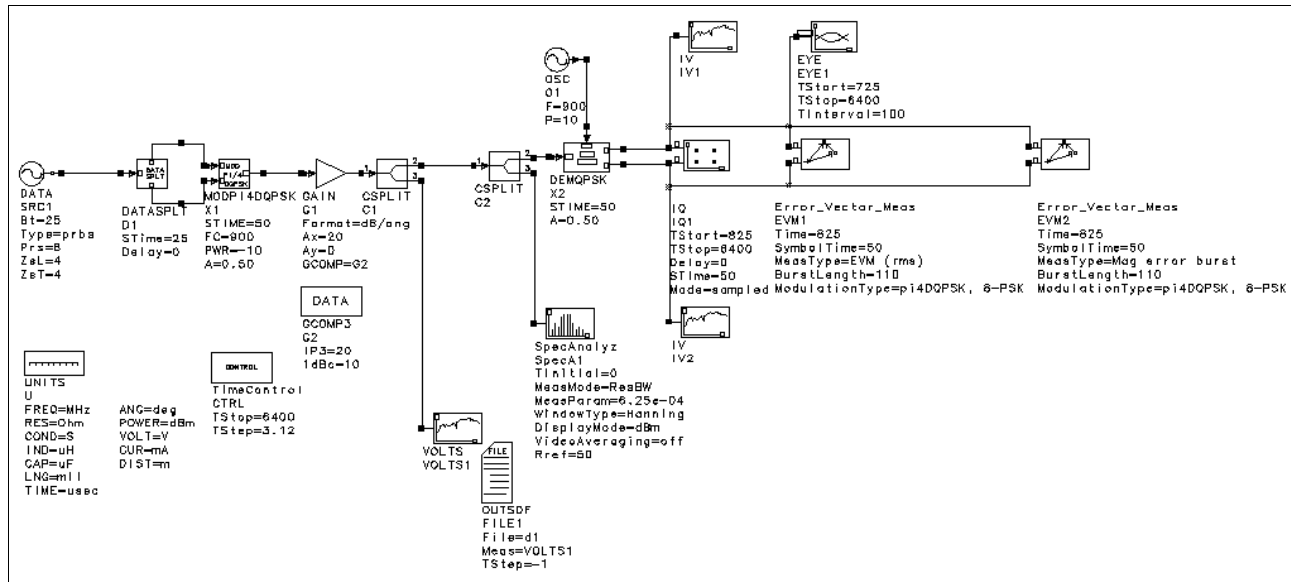
The demonstration in this section shows amplifier compression effects being evaluated with distortion data obtained by simulation and measurement methods using OmniSys/CDS and a VSA. This demonstration assumes the VSA is an HP 89440A, a LAN connection between the workstation and VSA, and that the RF source output is jumpered to the measurement receiver input. While the VSA X-Window can be displayed either by a *telnet* session or by use of an AEL scripted menu item, the use of the menu is shown. (Refer to the section *AEL File for Custom Menu* at the end of this product note for a description of the custom menu AEL file and installation instructions.)

This demonstration is run from a project directory called *hp89440\_prj*, which contains two OmniSys/CDS test benches, a *project\_state.ael* file, and two VSA *state.sta* files, as well as other items. The simulator test benches and VSA state files are described, then the demonstration is described. If the project file is not available, the simulator test benches and VSA state files can be entered as described. To obtain a copy of the project use either *anonymous ftp* from [hpeesof.external.hp.com/distribution/app\\_note/hp89440\\_prj.tar.Z](http://hpeesof.external.hp.com/distribution/app_note/hp89440_prj.tar.Z), or from the World-Wide Web at <http://www.hp.com/go/hpeesof>.



## Test Bench Description

The first test bench needed in the new *hp89440\_prj* is called *pi4write\_tb* and is shown in Figure 7. This bench creates an arbitrary signal that is used here in an ideal OmniSys/CDS amplifier (it will be distorted later in this example); this amplifier could be an actual amplifier inserted between the VSA source output and measurement input. The amplifier output, whether virtual or real, is measured in both OmniSys/CDS and in the VSA.



**Figure 7. *pi4write\_tb* Test Bench**

This test bench consists of a pseudo random bit stream data generator source DATA (*Type=prbs*) feeding a data splitter, which then drives a  $\pi / 4$  DQPSK modulator at 900 MHz. The modulator output enters an ideal amplifier where the output is split three ways: into a spectrum analyzer; into a VOLTS measurement that feeds an OUTSDF file; and, into a receiver beginning with an ideal, distortionless QPSK demodulator (DEMQPSK).

The demodulator is referenced to a 900 MHz LO (OSC) so that baseband I and Q signals are available for IQ diagram, EYE diagram, EVM rms, and EVM magnitude error burst measurements. The IQ diagram actually displayed uses the IV1 and IV2 measurements in an AVB measurement to show a continuous ball-of-yarn constellation rather than the sampled output available from the IQ1 measurement. This sampled measurement is necessary to confirm the simulation settings for TStart, TStop, and STime. Once the correct settings are observed, they are manually transferred to the EYE and EVM measurements.

In Figure 7 observe that TStep in the time control item is actually 3.125  $\mu$ sec (the final digit is not visible on the test bench). This yields 2048 time points in the simulation and the SDF file, or 16 samples per symbol. Since the prbs generator is set for  $2^8 = 256$  bits, there are 128 symbols used in the simulation. This sequence is intentionally set somewhat short to keep the simulation times short, and to produce a constellation that is not completely filled in. A partially-completed constellation can be recognized and distinguished from among similar ones. In this manner, similar constellations are visible in OmniSys/CDS and on the VSA.

---

**Note** If the DATA source Type=*random* is in OmniSys/CDS, then a new and arbitrary constellation rotation position and pattern occurs with every re-simulation. When this random data is displayed in the VSA, the pattern is preserved but the rotational position is not. The rotational position could be re-established if the VSA delay time is adjusted after each simulation.

Because  $\pi/4$  DQPSK is a differential coding scheme, the constellation has no preferred rotational position. However, OmniSys/CDS will always produce the same rotational position because the DATA source Type=*prbs*, which always generates the same bit sequence. In order to provide a more effective demonstration, the VSA is synchronized to OmniSys/CDS. A synchronization word is generated by the use of a Zero Substitution Length sequence in the prbs data generator ( $ZsL=4$ ,  $ZsT=4$ ). The VSA is synchronized to the pattern 0000, producing a display on the VSA that has the same orientation as the OmniSys/CDS plot.

Distortion can be added to the amplifier by adding a GCOMP3 modifier set for IP3=20 dBm and 1dBc=10 dBm. Because the signal applied to the amplifier is -10 dBm and its gain is 20 dB, the output signal will be approximately +10 dBm, and the amplifier will be very slightly compressed. In this example, the result of the compression will be measured in both OmniSys/CDS and the VSA.

The second test bench needed is called *pi4read\_tb*; it is shown in Figure 8. This bench takes an output signal from the VSA and uses OmniSys/CDS to measure certain signal characteristics. In this way, system measurement results are included in the simulation, and the simulation itself becomes virtual hardware. This shows that distortion effects from virtual or real hardware as measured by the VSA are confirmed by OmniSys/CDS.

This bench reads a VSA output file named *trace1.dat* into an OmniSys/CDS QAM\_F source. The data consists of IQ and carrier information and is provided to a DEMQAM item where it is demodulated from 900 MHz. The baseband IQ data is then supplied to a continuous and sampled IQ measurement, an EYE display, an EVM rms, and an EVM magnitude error burst measurement. The sampled IQ measurement is used for the purposes of confirming timing settings as in the first test bench. Note that the TimeControl (TStep and TStop) settings are based on the VSA SDF file information available at simulation time in the simulator dialog box.

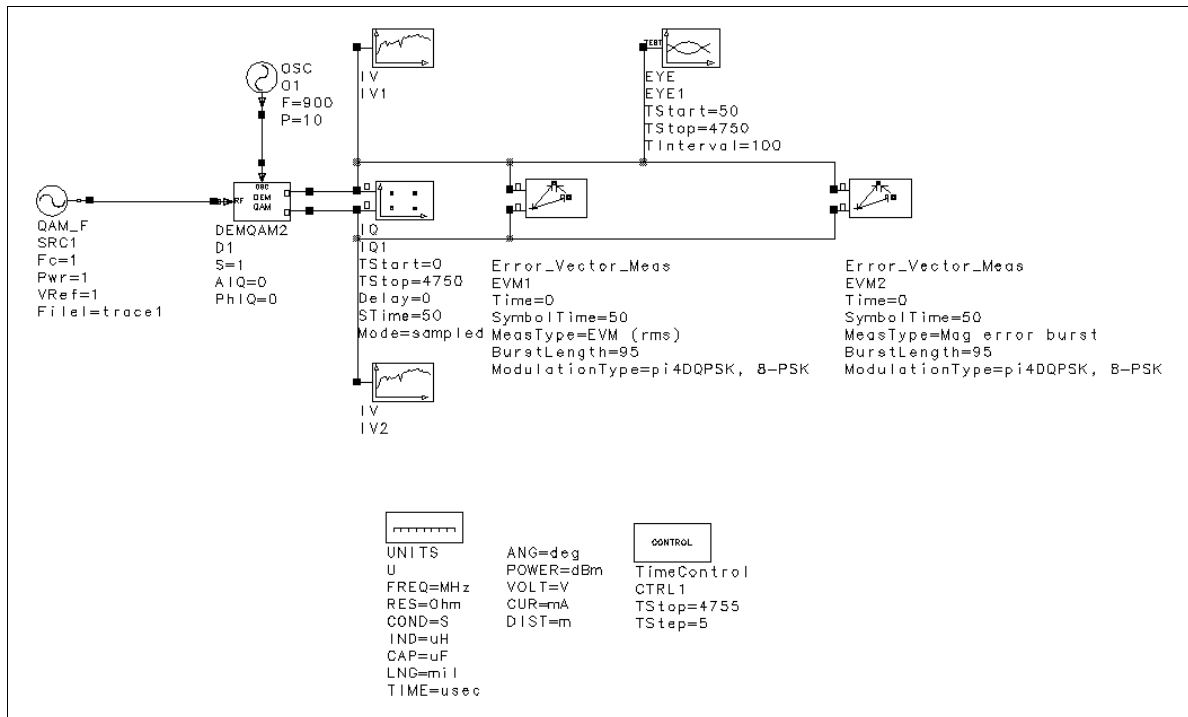


Figure 8. *pi4read\_tb* Test Bench

## VSA State Descriptions

The VSA state that corresponds to *pi4writer.sta* (mentioned earlier) can be created by entering the information shown in Figures 9 and 10 into the VSA, and saving the resulting *state.sta* file from the VSA /data directory as *pi4writer.sta* in the *hp89440\_prj* data directory.

INPUT/SOURCE STATE				
<b>Range/ Input</b>	-- Channel 1 --		-- Channel 2 --	
	Status	on	Status	-0 dBm
	Range	20 dBm	Range	-0 dBm
	Input 2	50 Ohm	Autorange	off
			Input 2	50 Ohm
			IM imped	50 Ohm
			Coupling	AC
			Alias LPF	in
<b>Trigger</b>	Type	int source	Level	0 V
	Slope	+		
	Ch1 delay	1.1562 ms	Ch2 delay	0 s
	Ext arm	off	Arm level	0 V
	Arm slope	above	Arm delay	0 s
	Holdoff	off	Hld delay	0 s
<b>Source</b>	Status	on	Level	0 dBVpk
	Type	arbitrary	Offs freq	0 Hz
	Output 2	50 Ohm		

**Figure 9. VSA Input/Source State Corresponding to *pi4writer.sta***

MEASUREMENT STATE				
<b>Inst Mode</b>	Digital Demodulation	Channels	2	
	Meas from input	Receiver	RF 1800 MHz	
	Format	pi/4 DQPSK	Rate	20 kHz
	Alpha	0.5	Meas filter	root RC
	Normalize	on	Ref filter	RC
	Clk Adj	0 sym		
<b>Freq</b>	Span	50 kHz	Center	900 MHz
	<b>Bw/Win</b>	Rbw	1 kHz	Window
<b>Time</b>	Puls-srch	off	Sync-srch	on
	Pts/sym	10	Srch	5.2 ms
	Result	96 sym	Sync offset	4 sym
	Sync word	0000		
<b>Average</b>	Status	off	Num averages	10
	Type	rms expo	Repeat avg	off
	Overlap	0 %	Fast avg	off

**Figure 10. VSA Measurement State Corresponding to *pi4writer.sta***

In addition to setting the instrument mode, center frequency, and span information, the measurement state is set to control delay and sync information, as well as measurement and reference filter configuration. Note that the Sync offset and Sync word correspond to ZsT and ZsL, respectively.

Recall that the symbol time is 50  $\mu$ sec so that 128 symbols (or a 6.4 msec span) is generated in OmniSys/CDS. Since the VSA cannot be run over the start and stop times of a data stream, only 5.2 msec is searched (104 symbols) to ensure continuity from end to beginning. 96 symbols are actually used, and 4 symbols are used for offset and 4 symbols are used for sync. The Channel 1 delay of 1.1562 msec (23-1/8 symbols) is used to achieve constellation rotation in the VSA consistent with OmniSys/CDS.

The VSA state that corresponds to *pi4writet.sta* can be created by entering the information shown in Figures 11 and 12 into the VSA, and saving the resulting *state.sta* file available in its */data* directory as *pi4writet.sta* in the *hp89440\_prj* data directory.

In this state the VSA is used as a spectrum analyzer, so only the center frequency and span information need to be set. The source power level is also set to correspond to the amplifier output power level. Recall that units of dBm cannot be set from the front panel (20 dBVpk equals 10 dBm in a 50-ohm system); however, dBm units can be entered from the SCPI command line.

INPUT/SOURCE STATE			
<b>Range/</b>	-- Channel 1 --	-- Channel 2 --	
<b>Input</b>	Status on	Status on	
	Range 10 dBm	Range -0 dBm	
	Input Z 50 Ohm	Autorange off	
		Input Z 50 Ohm	
		IM imped 50 Ohm	
		Coupling AC	
		Alias LPF in	
<b>Trigger</b>	Type int source	Level 0 V	
	Slope +		
	Ch1 delay 1 ms	Ch2 delay 0 s	
	Ext arm off	Arm level 0 V	
	Arm slope above	Arm delay 0 s	
	Holdoff off	Hld delay 0 s	
<b>Source</b>	Status on	Level 0 dBVpk	
	Type arbitrary	Offs freq 0 Hz	
	Output Z 50 Ohm		

**Figure 11. VSA Input/Source State Corresponding to *pi4writet.sta***

MEASUREMENT STATE			
<b>Inst</b>	Vector	Channels	2
<b>Mode</b>	Meas from input	Receiver	RF 1800 MHz
<b>Freq</b>	Start 899.95 MHz	Center	900 MHz
	Stop 900.05 MHz	Span	100 kHz
<b>Res Bw/</b>	Rbw 1 kHz	Rbw coupling	auto
<b>Window</b>	Rbw mode 1-3-10	Time len	3.8203 ms
	Window flat top	Num freq pts	401
<b>MainTime</b>	Ch1 delay 1 ms	Width	3.8203 ms
	Ch2 delay 0 s	Gate	off
<b>GateTime</b>	Ch1 delay 0 s	Width	132.81 us
	Ch2 delay 0 s		
<b>Average</b>	Status off	Num averages	10
	Type rms expo	Repeat avg	off
	Overlap 0 %	Fast avg	off

**Figure 12. VSA Measurement State Corresponding to *pi4writet.sta***

## Demonstration

Once the OmniSys/CDS project and VSA state information files are available, open the demonstration by starting OmniSys/CDS:

- If OmniSys/CDS is started from within the project, the *project\_state.ael* file is able to read in all project files and to restore simulation windows and locations.
- If OmniSys/CDS is started from outside the project, first use the Main PDE Window to access the *Project / Open / hp89440\_prj* menu command and then use *Option / Playback Macro* to run the *project\_state.ael* file. This restores the project state in the same way as starting OmniSys/CDS from within the project.

(See the *Demonstration Setup* section for information on how to obtain a copy of the *hp89440\_prj*.)

Once OmniSys/CDS is open, the HP89440A menu (as part of the test bench) is used to open the VSA virtual display as an X-Window. (Refer to sections at the end of this document for information regarding the HP89440A custom menu, and for information regarding the color mapping configuration file.)

An *ftp* session is also established in a separate window that originates in the *hp89440\_prj/data* directory and is targeted to the *VSA/data* directory. All simulation and test items can be placed in the test bench; a schematic window is not necessary. Refer to the *Demonstration Setup* section for information regarding obtaining a copy of *hp89440\_prj*.

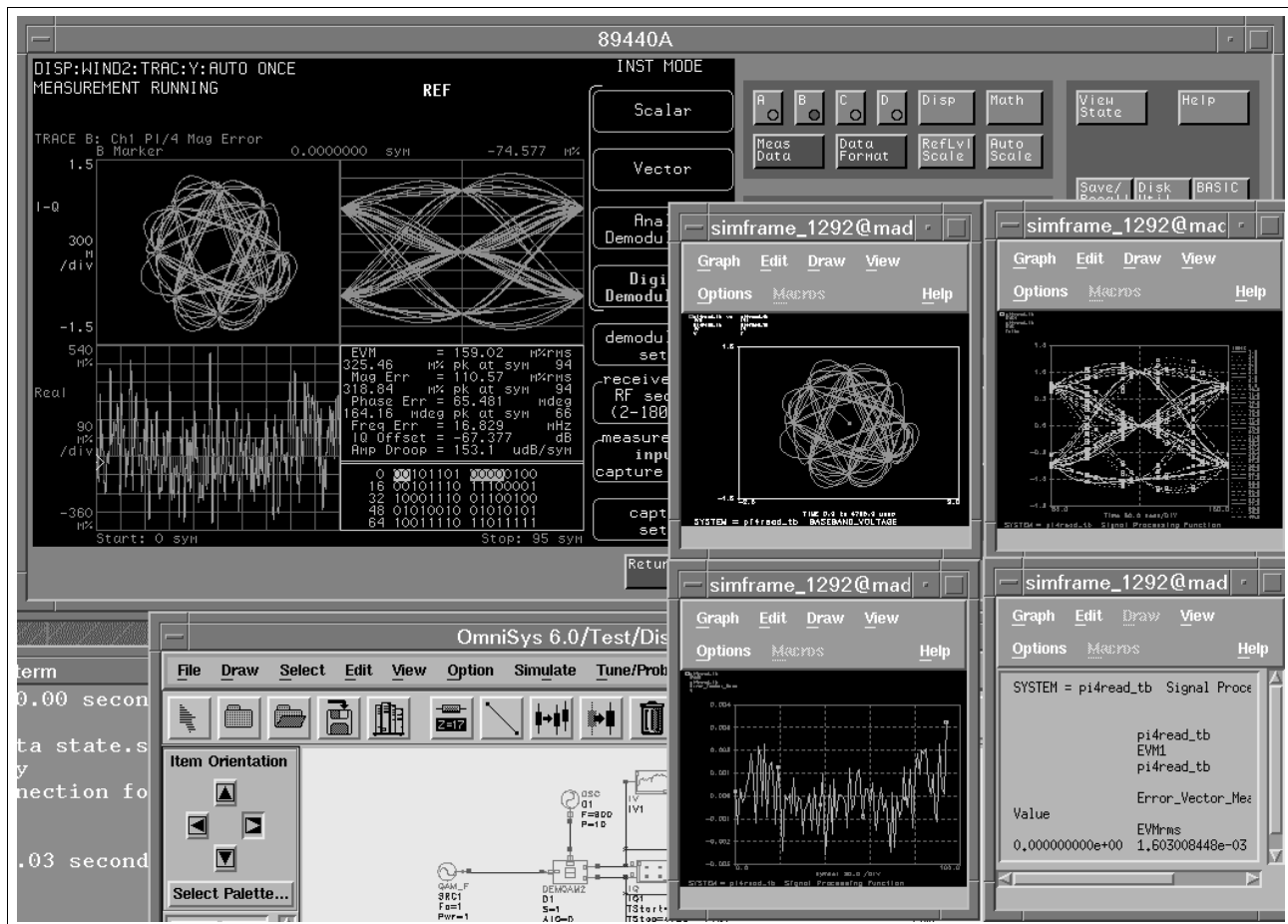


OmniSys/CDS and the VSA. The constellation, eye, and error vector magnitude, are all in excellent agreement. Even the rotational position of the constellations are identical. For EVM rms, OmniSys/CDS reports 0.05% (5.15 E-4), and the VSA reports 0.16% (161 milli-percent). Both numbers are extremely low, but OmniSys/CDS has the advantage, in that the simulation is noiseless and the VSA produces some (very low-level) noise.

**Demonstration—Part 2**

1. Use the OmniSys/CDS menu commands  
**Graph / Close (<Ctrl>-c) / all four pi4writer.gra**  
**File / Read / pi4read.tb**  
**Results / Load Graph / pi4read.gra**
2. In the *ftp* window, enter **ftp > get trace1.dat**
3. Use the OmniSys/CDS menu command **Simulate / Analyze (F7) / pi4read.tb**
4. Compare simulated and measured data. The workstation should resemble Figure 14.

Here, a signal with embedded noise is taken from the VSA measurement receiver output and given to OmniSys/CDS by means of a *trace1.dat* file. The simulator reports readings identical to the VSA at a low 0.16 percent EVM rms.



**Figure 14. VSA Session Output and OmniSys/CDS Session Output Corresponding to Demonstration—Part 2**



**Demonstration—Part 3**

1. Use the OmniSys/CDS menu commands

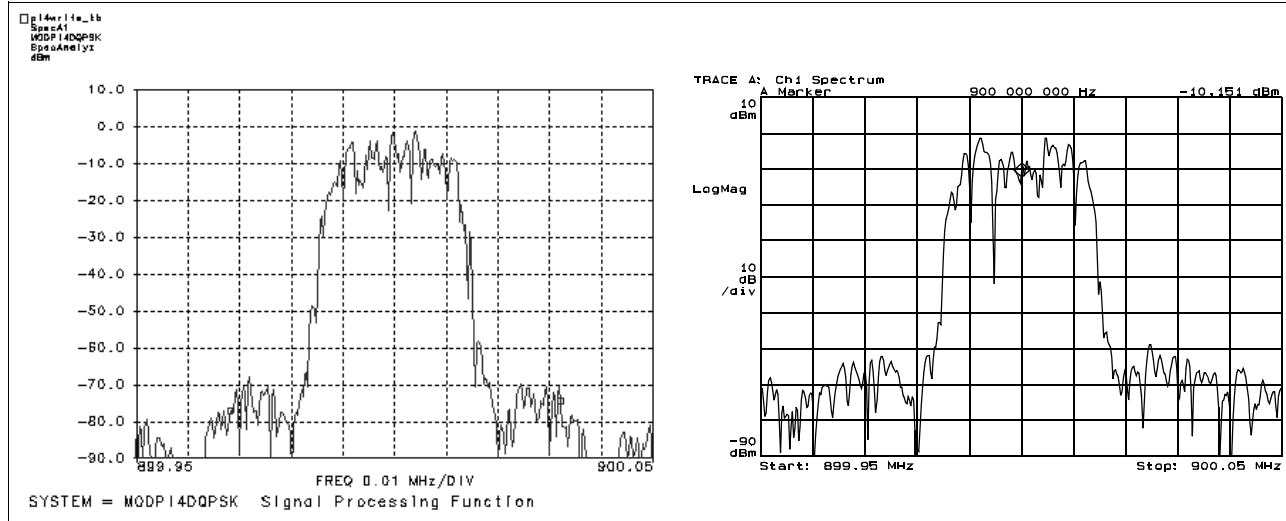
**Graph / Close (<Cntrl>-c) / all four pi4read.gra**

**File / Read / pi4write\_tb**

**Results / Load Graph / pi4writet.gra**

2. In the *ftp* window, enter **ftp > put pi4writet.sta state.sta**
3. Use the OmniSys/CDS menu command **Simulate** (if necessary)
4. Compare simulated and measured data; see Figure 15.

As shown in Figure 15, the same undistorted signal is being displayed by the simulator and the VSA. Note that the first set of sidebands is  $-60\text{dBc}$ . The second set of sidebands, visible in the simulation, are below the noise floor of the VSA at  $-70\text{ dBc}$ . Note also that the power measurements are nearly identical.

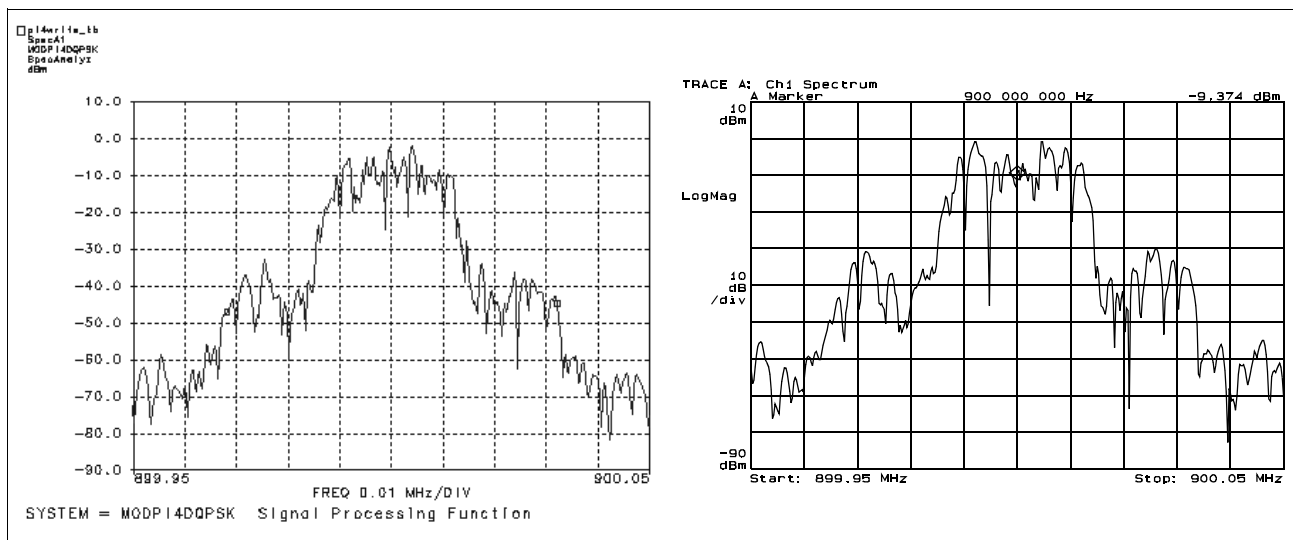


**Figure 15. OmniSys/CDS Output (left) and VSA Output (right)  
Corresponding to Demonstration—Part 3**

**Demonstration—Part 4**

1. Distort the amplifier by **adding a GCOMP3 element** associated with the GAIN G1 amplifier
2. Use the OmniSys/CDS menu command **Simulate / Analyze (F7) / pi4write\_tb**
3. In the *ftp* window, enter **ftp > put d1.dat**
4. Compare simulated and measured data; see Figure 16.

As shown in Figure 16, a very small amount of distortion is introduced by compressing the amplifier. Not only is the first set of sideband levels similar (-30 dBc), but so is the second set (-60 dBc). An OmniSys/CDS amplifier is used here—the use of a real amplifier would give similar results.



**Figure 16. OmniSys/CDS Output (left) and VSA Output (right)  
Corresponding to Demonstration—Part 4**

**Demonstration—Part 5**

1. Use the OmniSys/CDS menu commands

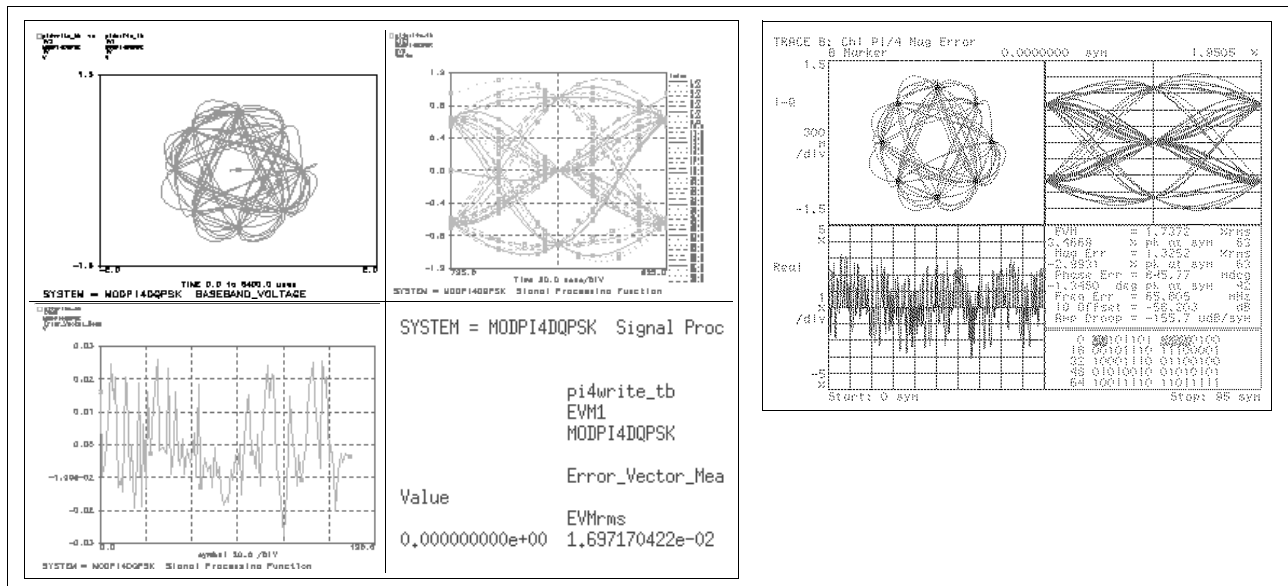
**Graph / Close (<Ctrl>-c) / pi4writet.gra**

**Results / Load / pi4writer.gra**

2. In the *ftp* window, enter **ftp > put pi4writer.sta state.sta**
3. Compare simulated and measured data; see Figure 17.

With the same amount of amplifier distortion as in the previous demonstration, the EVM rms is now 1.7 percent, measured with the simulator or the VSA.

This demonstration shows that hardware effects can be included in a simulation, or simulation effects can be included in the hardware. And, emulation of virtual hardware then arbitrary signal generation for RF system stimulation are shown.



**Figure 17. OmniSys/CDS Output (left) and VSA Output (right) Corresponding to Demonstration—Part 5**

## AEL File for Custom Menu

The AEL (script) file for adding a custom HP89440A menu item to OmniSys/CDS is shown below. A copy of this text can be found in the project *hp89440\_prj*.

```
// Install this file in your $HOME/ael/omnisys. (CDS also uses
// $HOME/ael/omnisys. Create if necessary. Configure this file by
// changing mach and host to your site. Either nslookup or /etc/hosts
// or both. If you are running hubless, make the 89440 address to be only
// one digit different from the computer (use the same subnet). Create a
// .simframe_omni and .simframe_comms in your $HOME (if necessary). Add
// to each .simframe_xxx file the line: USER_AEL=hp89440 (this file). This
// ael file then causes an atf file to be compiled. Not necessary to reboot,
// only restart SIV. New menu will appear in schematic (SCHEM_WIN)or test
// bench (TESTS_WIN>window.
#voc(CmdOp) //Vocabulary specification
fputs(stderr,"HP89440A AEL is loading.");

decl mach = "hp89440.wlv.hp.com"; // test instrument name (as from nslookup)
decl host = "15.13.16.81"; // your host machine IP address (nslookup)

decl inst = create_server( "telnet","HP89440A session","",mach, 1 );

push_message_handler( print_sim_message, inst );

defun turn_on()
{
    system(strcat("xhost +",mach));
    send_server_data( strcat("\n\nsyst:comm:lan:xwin:host '",host,
        "'\n\nsyst:comm:lan:xwin on\n"), inst );
}

defun turn_off()
{
    send_server_data( "\n\nsyst:comm:lan:xwin off\n", inst );
    send_server_kill(inst);
    system(strcat("xhost -",mach));
}

/*****
/* open_window function */
*****/

defun open_window (winType)
{
    decl newWinCreated;

    newWinCreated = create_window (winType);
    if (newWinCreated)
    {
        if (winType == TESTS_WIN)
        {
            add_menu("on", "turn_on");
            add_menu("off", "turn_off" );
            set_user_menu_label( "HP89440A" );
        }
        // add user menu for the winType
    }
}
}
```

## Color Mapping Configuration File

If the color map conflicts between OmniSys/CDS and the VSA, place a copy of \$EESOF\_DIR/config/eecolor.cfg in your \$SHOME directory, name it .eecolor, point to it in .eesof, and modify it similar to the file hp89440.eecolor, which is shown below. .eesof is a copy of \$EESOF\_DIR/config/eesof.cfg and must be placed in your \$SHOME directory. A copy of this text can be found in the project hp89440\_prj.

```
#
# =====
# EEs of Color Definition File:
# =====
#
#   R   G   B   :   Color Name   :   Plotter Pen #
#   --- --- --- :   -----
#   0   0   0   :   black       :   1
#  255  0   0   :   red         :   2
#   0  255  0   :   green      :   3
#   0   0  255  :   blue       :   4
#  255 255  0   :   yellow     :   5
#  255  0  255  :   magenta    :   6
#   0  255 255  :   cyan       :   7
#  255 255 255  :   white      :   8
#
# Either RGB values or standard X Color Names, or both, may be
# specified for color entries. Note, if both RGB and Color
# Name values are specified, the RGB values take precedence.
# =====
#
: black : 1
: red : 2
: yellow : 3
: green : 4
: cyan : 5
: blue : 6
: magenta : 7
: gray : 8
: white : 1
: medium blue : 1
: DarkSlateGrey : 1
: DimGrey : 1
: SlateGrey : 1
: MidnightBlue : 1
: NavyBlue : 1
: DarkSlateBlue : 1
: SlateBlue : 1
: MediumSlateBlue : 1
: medium blue : 1
: RoyalBlue : 1
: blue : 1
: DodgerBlue : 1
: SteelBlue : 1
: DarkGreen : 1
: DarkOliveGreen : 1
: SeaGreen : 1
: ForestGreen : 1
: OliveDrab : 1
: IndianRed : 1
: SaddleBrown : 1
: sienna : 1
: firebrick : 1
: brown : 1
: OrangeRed : 1
: red : 1
: DeepPink : 1
: maroon : 1
: MediumVioletRed : 1
: VioletRed : 1
: magenta : 1
```

## Evaluating System Distortion Effects by Simulation or Measurement Methods

```
: DarkOrchid : 1
: snow1 : 1
: snow2: 1
: AntiqueWhite1 : 1
: LemonChiffon1 : 1
: ivory1 : 1
: honeydew1 : 1
: LavenderBlush1 : 1
: LavenderBlush2 : 1
: MistyRose1 : 1
: MistyRose2 : 1
# : azure1 : 1
# : azure2 : 1
# : DodgerBlue2 : 1
# : SteelBlue1 : 1
# : DeepSkyBlue1 : 1
# : SkyBlue1 : 1
# : LightSkyBlue1 : 1
: SlateGray2 : 1
: LightSteelBlue1 : 1
: LightBlue1 : 1
: LightCyan1 : 1
: PaleTurquoise1 : 1
# : CadetBlue1 : 1
# : CadetBlue2 : 1
# : turquoise1 : 1
: cyan1 : 1
: DarkSlateGray1 : 1
: aquamarine2 : 1
: DarkSeaGreen1 : 1
: DarkSeaGreen2 : 1
# : SeaGreen1 : 1
# : PaleGreen1 : 1
# : SpringGreen1 : 1
# : green1: 1
# : chartreuse1: 1
: OliveDrab1: 1
: DarkOliveGreen1: 1
: khaki1: 1
: LightGoldenrod1: 1
: yellow1: 1
: gold1: 1
: goldenrod1: 1
# : sienna2: 1
# : wheat2: 1
: chocolate1: 1
# : brown1: 1
# : LightSalmon1: 1
# : orange1: 1
# : coral2: 1
: red2: 1
: DeepPink1: 1
: HotPink1: 1
: pink1: 1
: LightPink1: 1
: PaleVioletRed1 : 1
: maroon1 : 1
: VioletRed1 : 1
: magenta1 : 1
: orchid1 : 1
: plum1 : 1
: thistle1 : 1
: gray52 : 1
: gray62 : 1
: gray72 : 1
: gray82 : 1
```





For more information, contact a regional HP office listed below. Or check your telephone directory for a local HP sales office.

**United States**

800-452-4844

**Canada**

905-206-4725

**Europe (Amsterdam)**

European Marketing Center

P.O. Box 999

1180 AZ Amstelveen

The Netherlands

**Japan**

81-426-48-0237

**Latin America**

(Miami, Florida)

305-267-4245

**Australia/New Zealand**

13-1347 ext. 2902

**Asia Pacific (Hong Kong)**

8522-599-7070

**Data subject to change**

**© 1996 Hewlett-Packard Company**

**Printed in USA PN E4600-6**

**Rev. 6/96 2M**

**5964-3806E**